Metaphors and metaheuristics

A match made in hell

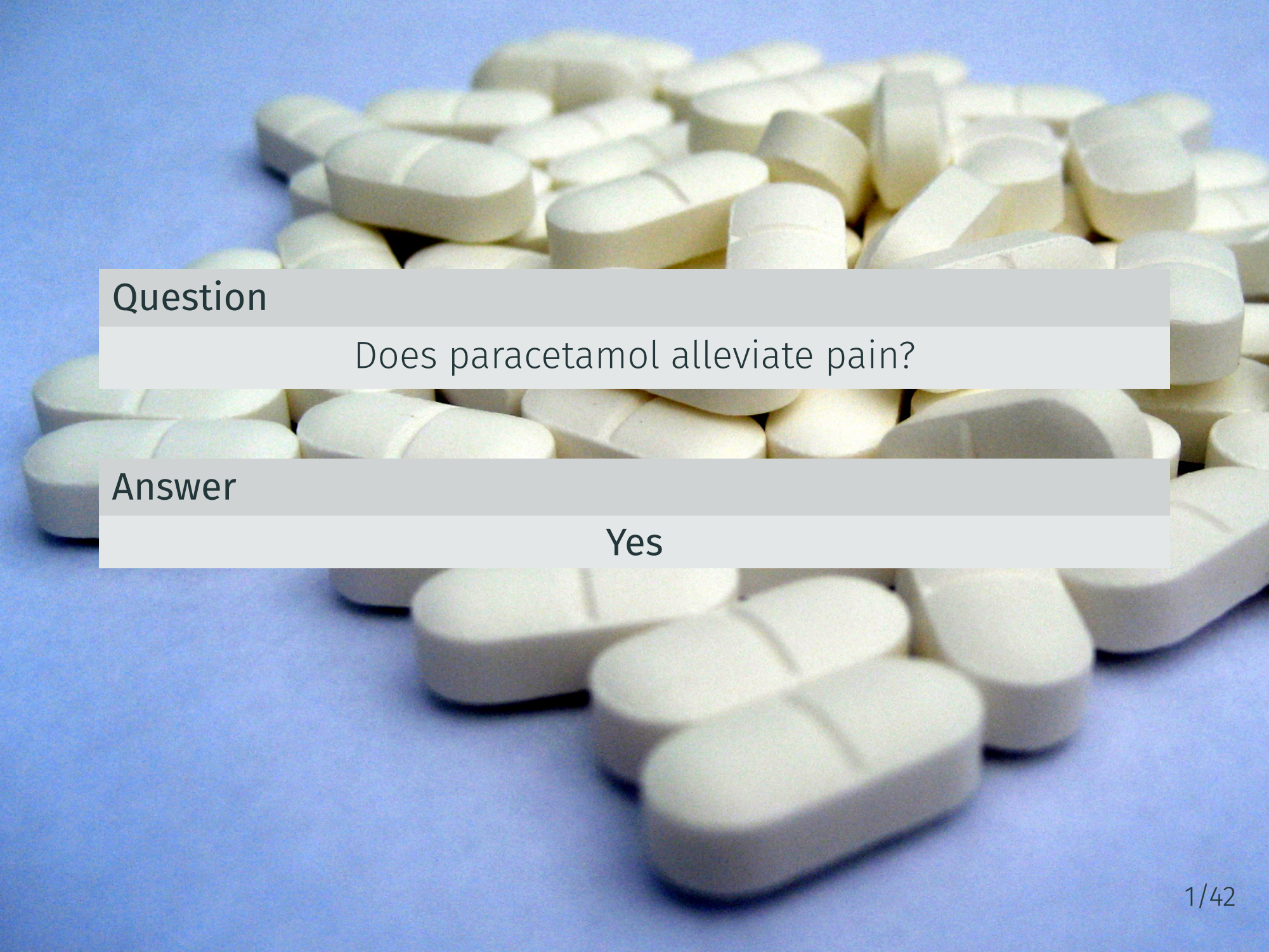
Kenneth Sörensen

Copenhagen – 23 April 2018

University of Antwerp
Operations Research Group
ANT/OR

Question

Does paracetamol alleviate pain?

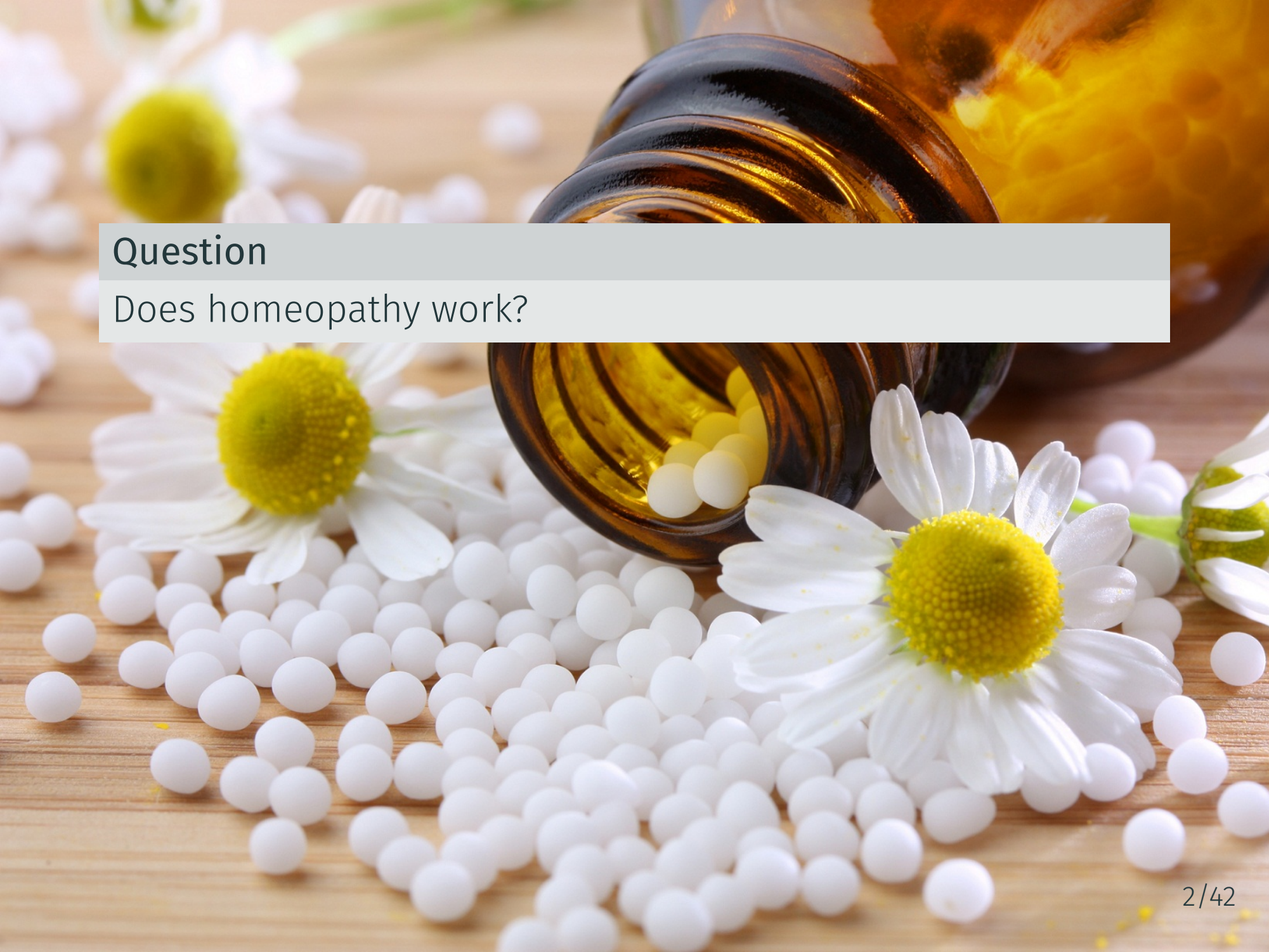


Question

Does paracetamol alleviate pain?

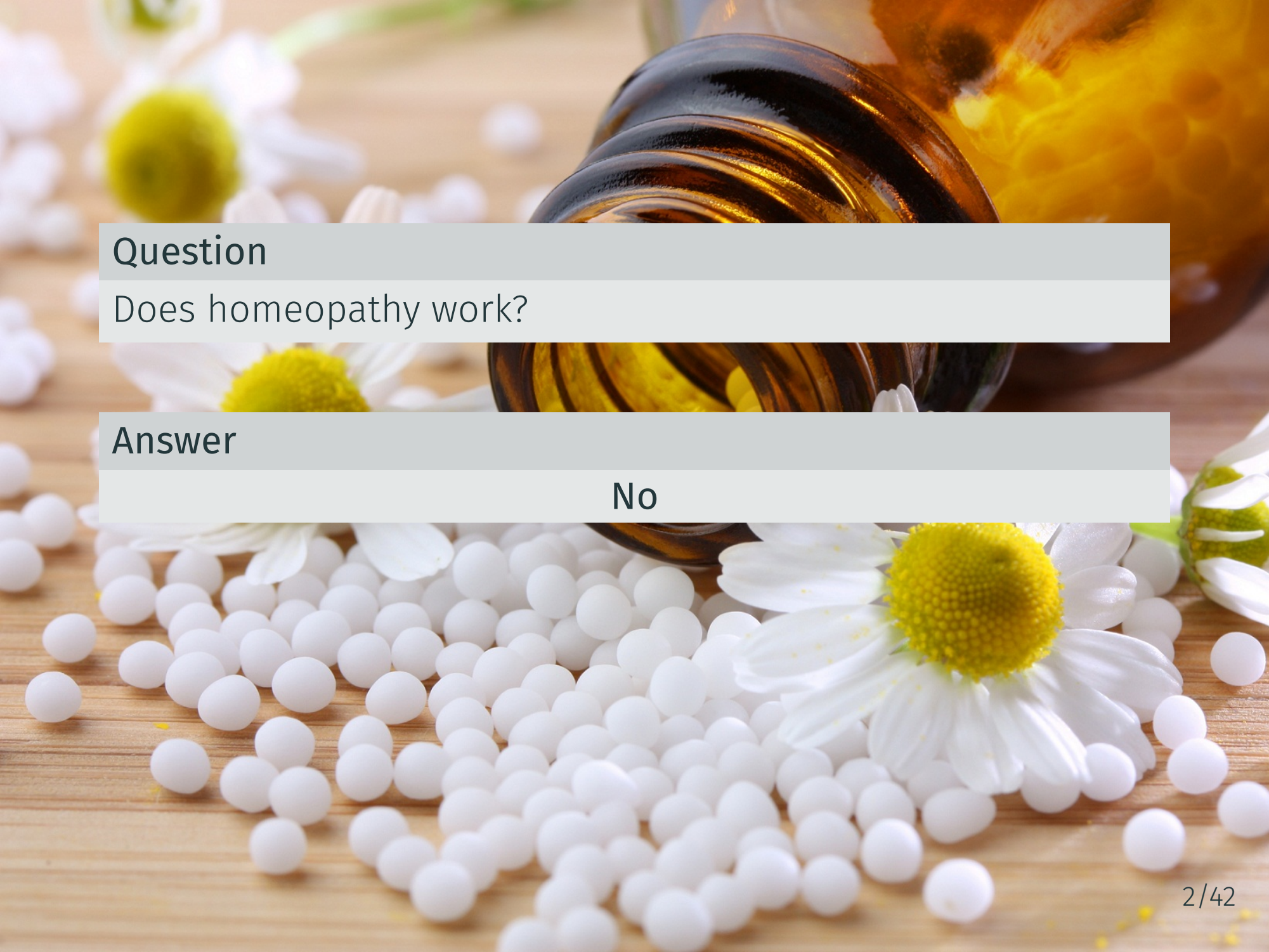
Answer

Yes



Question

Does homeopathy work?

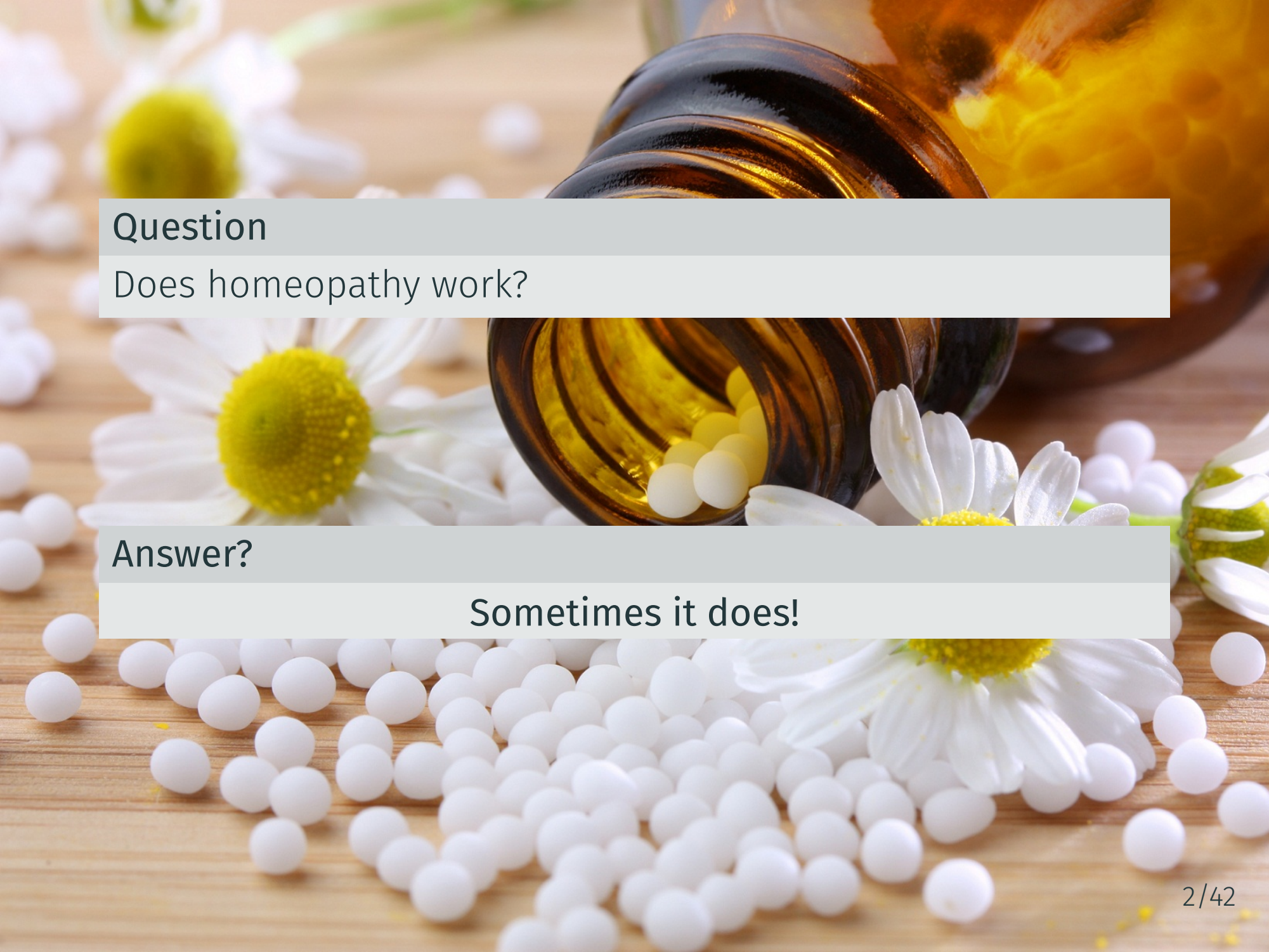


Question

Does homeopathy work?

Answer

No

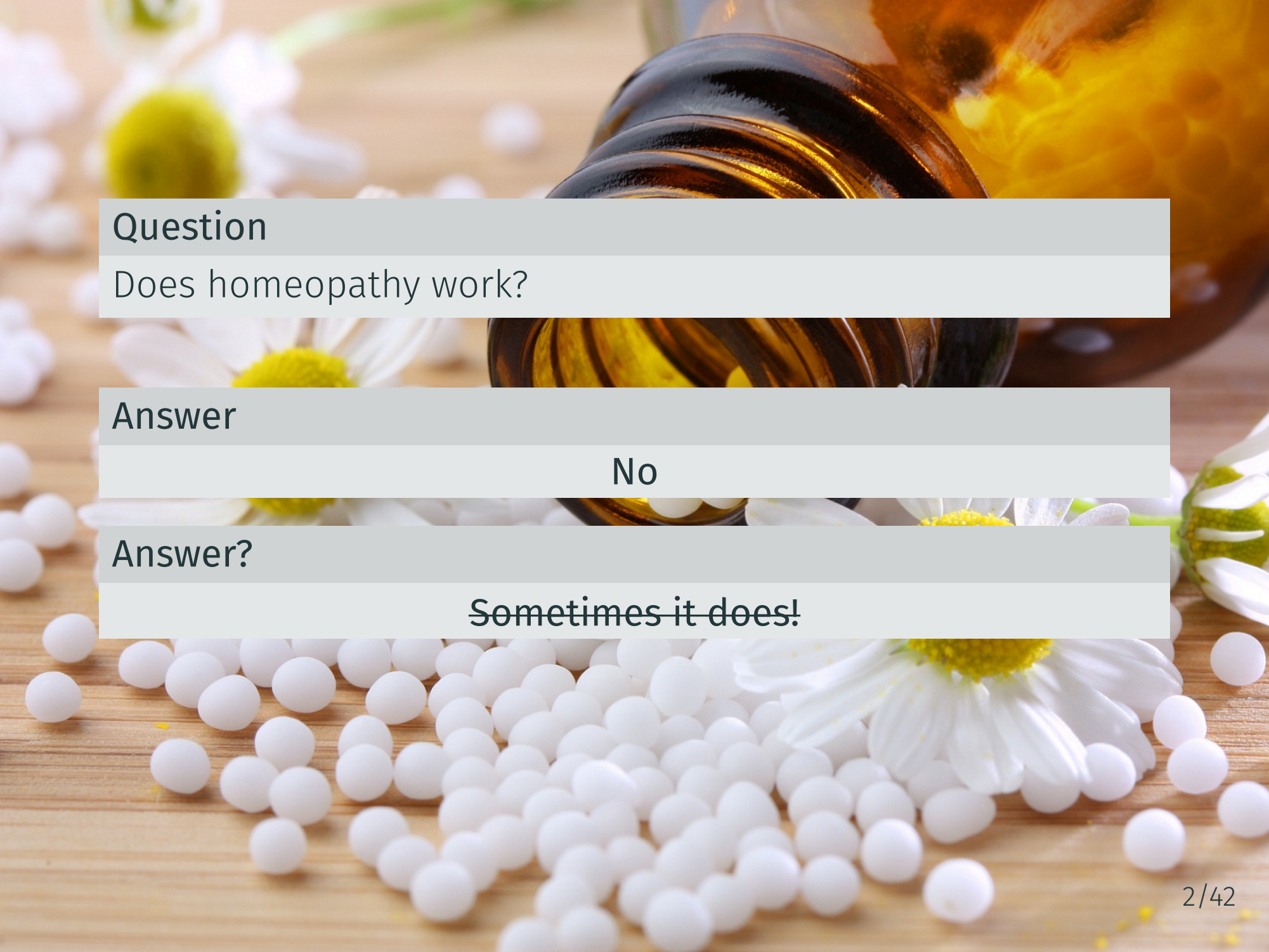


Question

Does homeopathy work?

Answer?

Sometimes it does!



Question

Does homeopathy work?

Answer

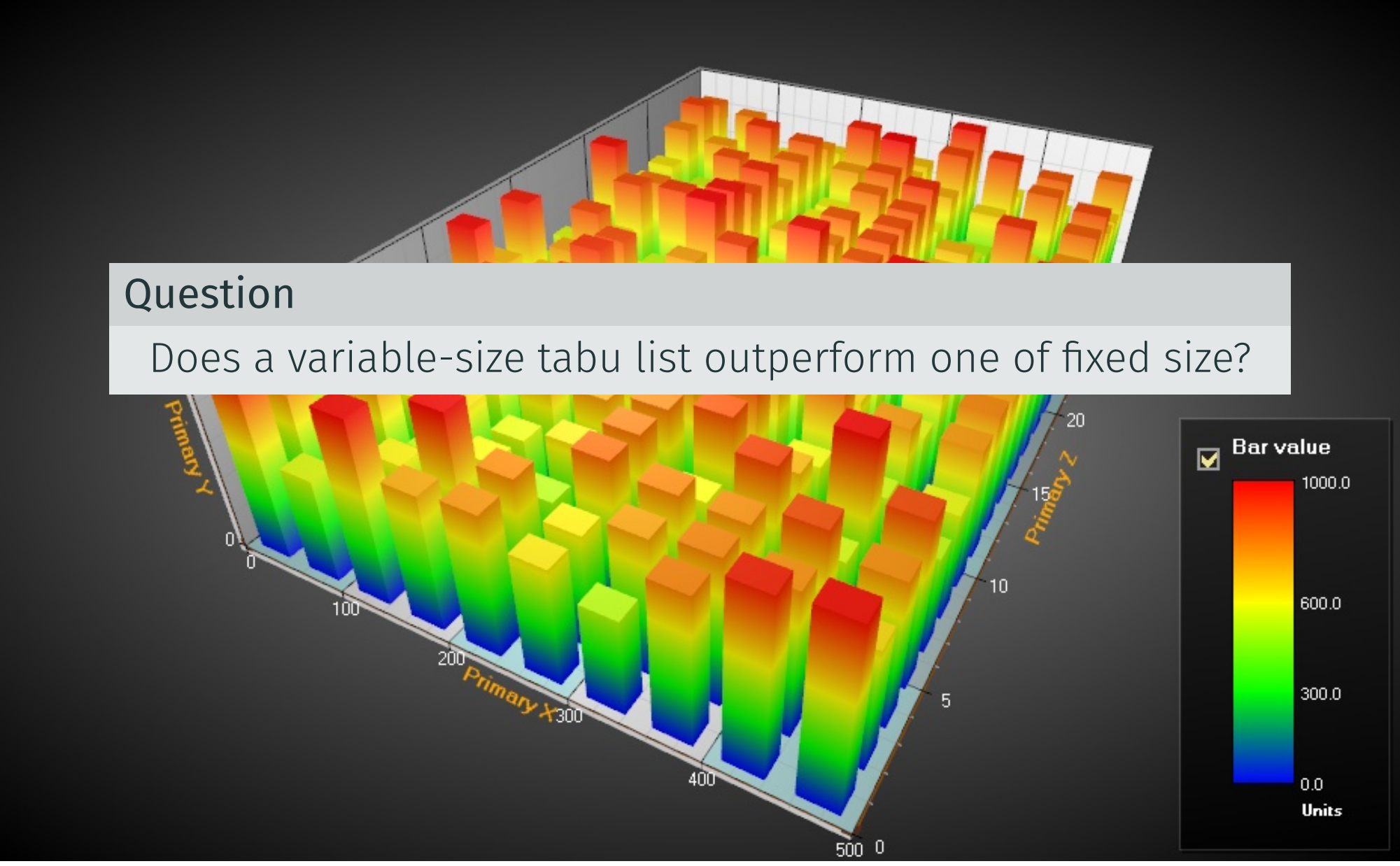
No

Answer?

~~Sometimes it does!~~

Question

Does a variable-size tabu list outperform one of fixed size?

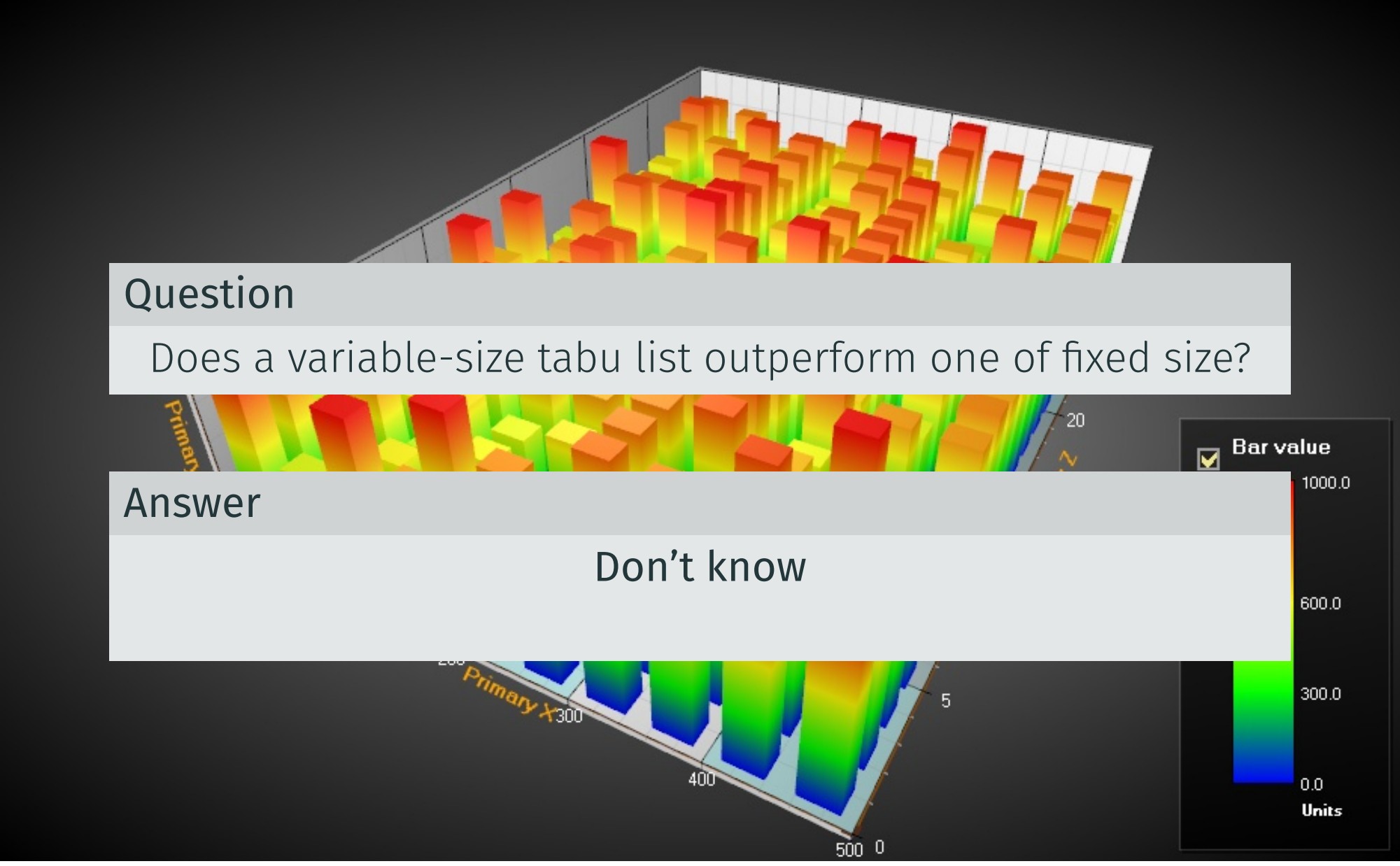


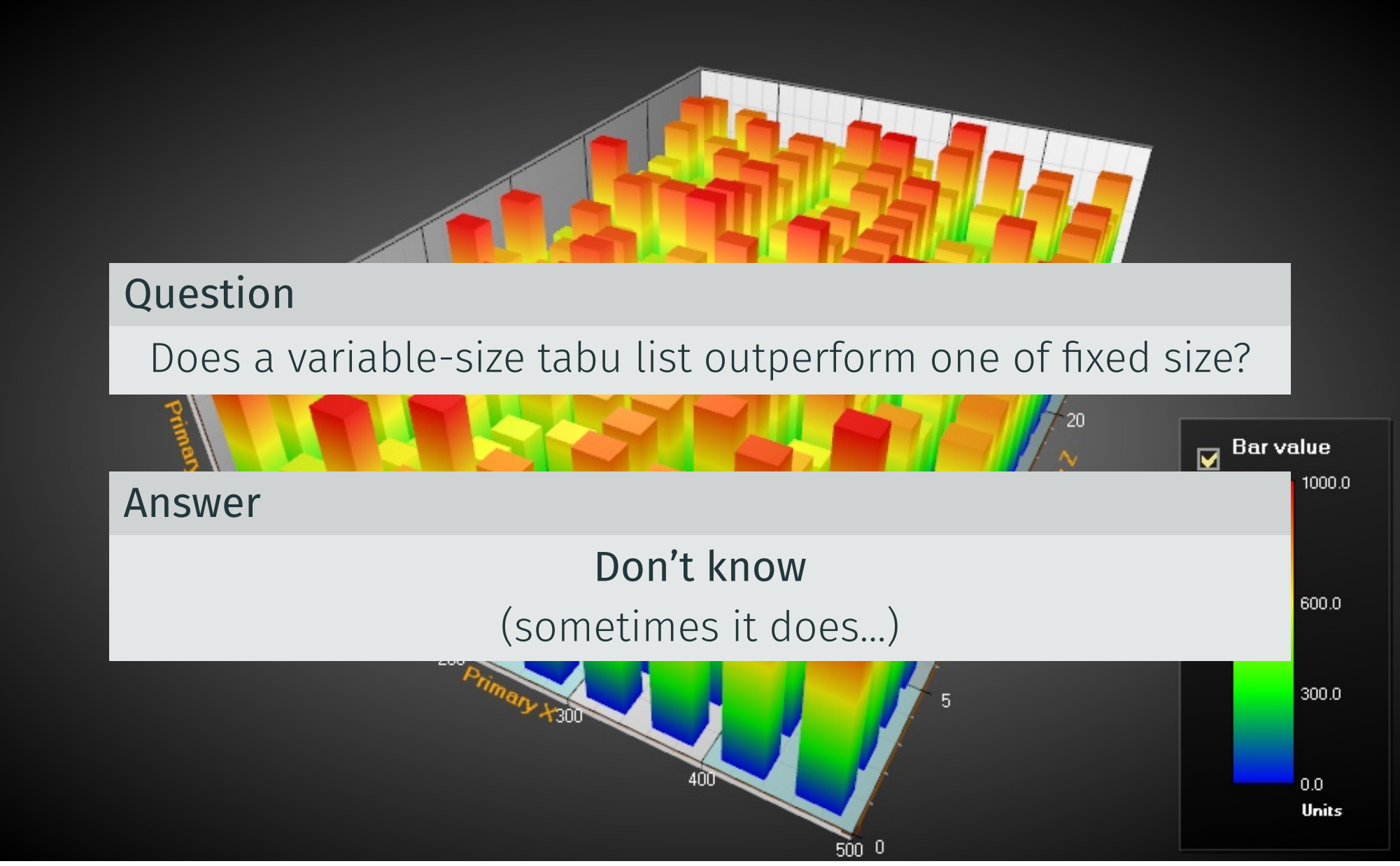
Question

Does a variable-size tabu list outperform one of fixed size?

Answer

Don’t know





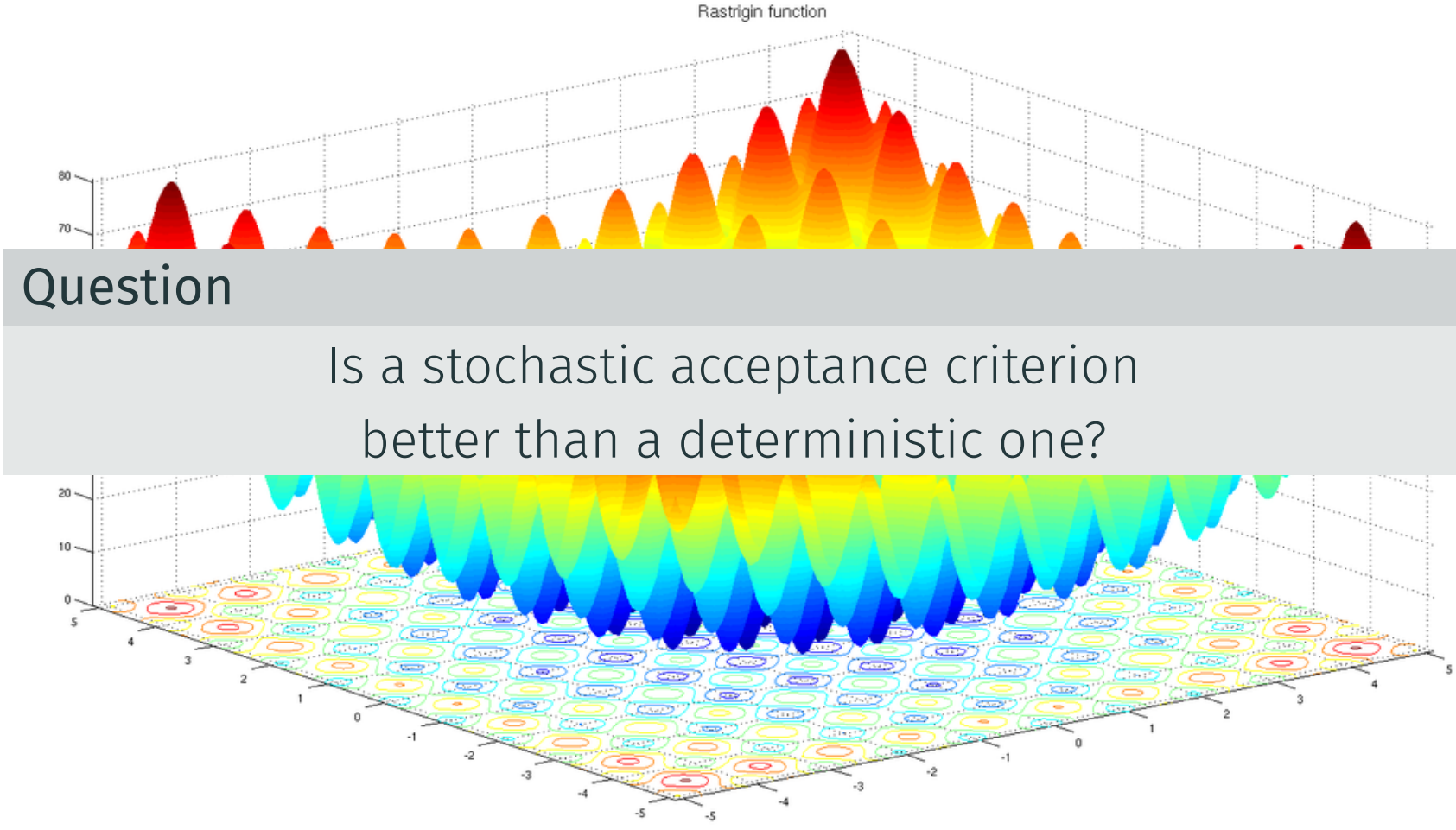
Question

Does a variable-size tabu list outperform one of fixed size?

Answer

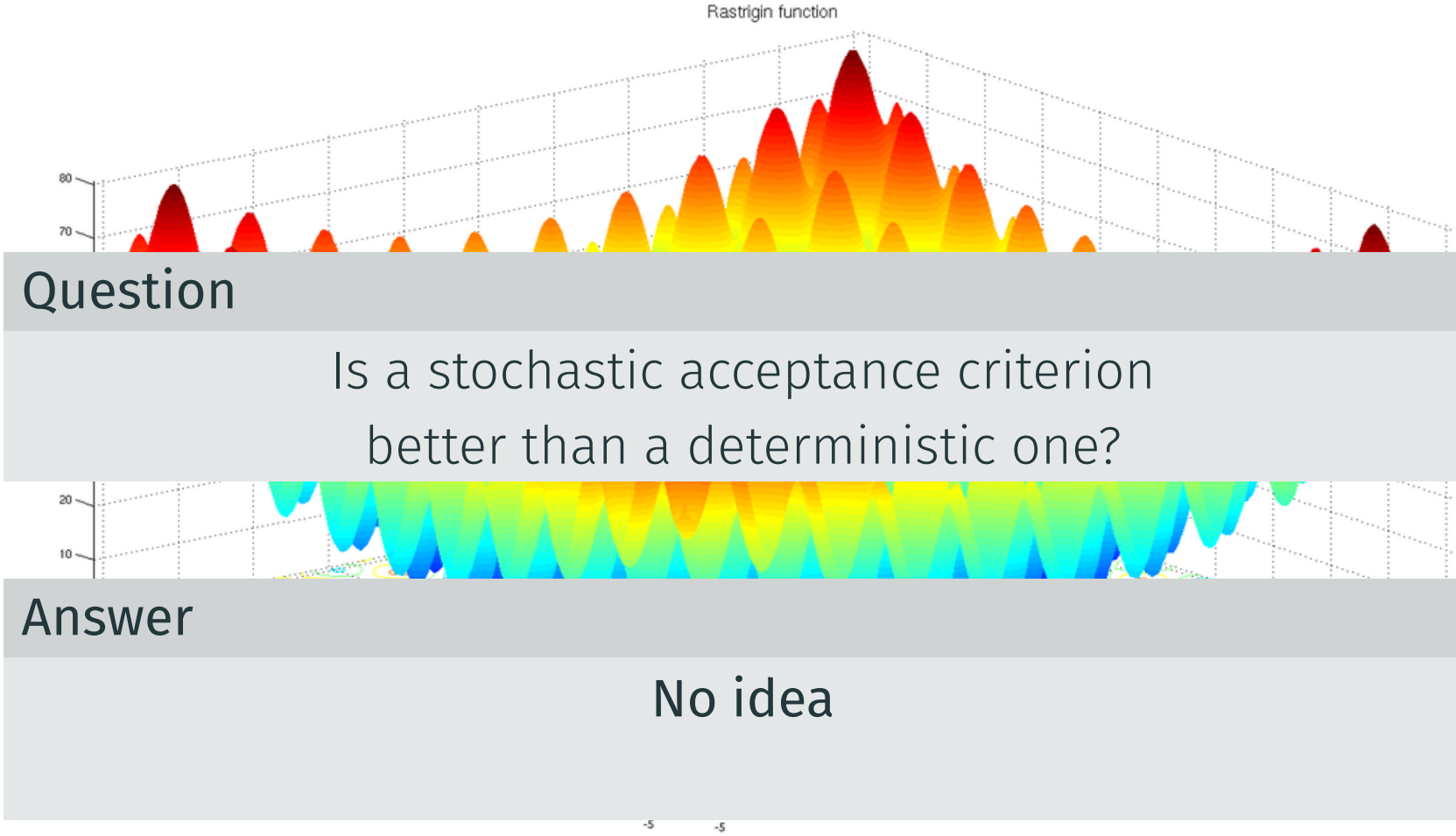
Don’t know

(sometimes it does...)



Question

Is a stochastic acceptance criterion
better than a deterministic one?

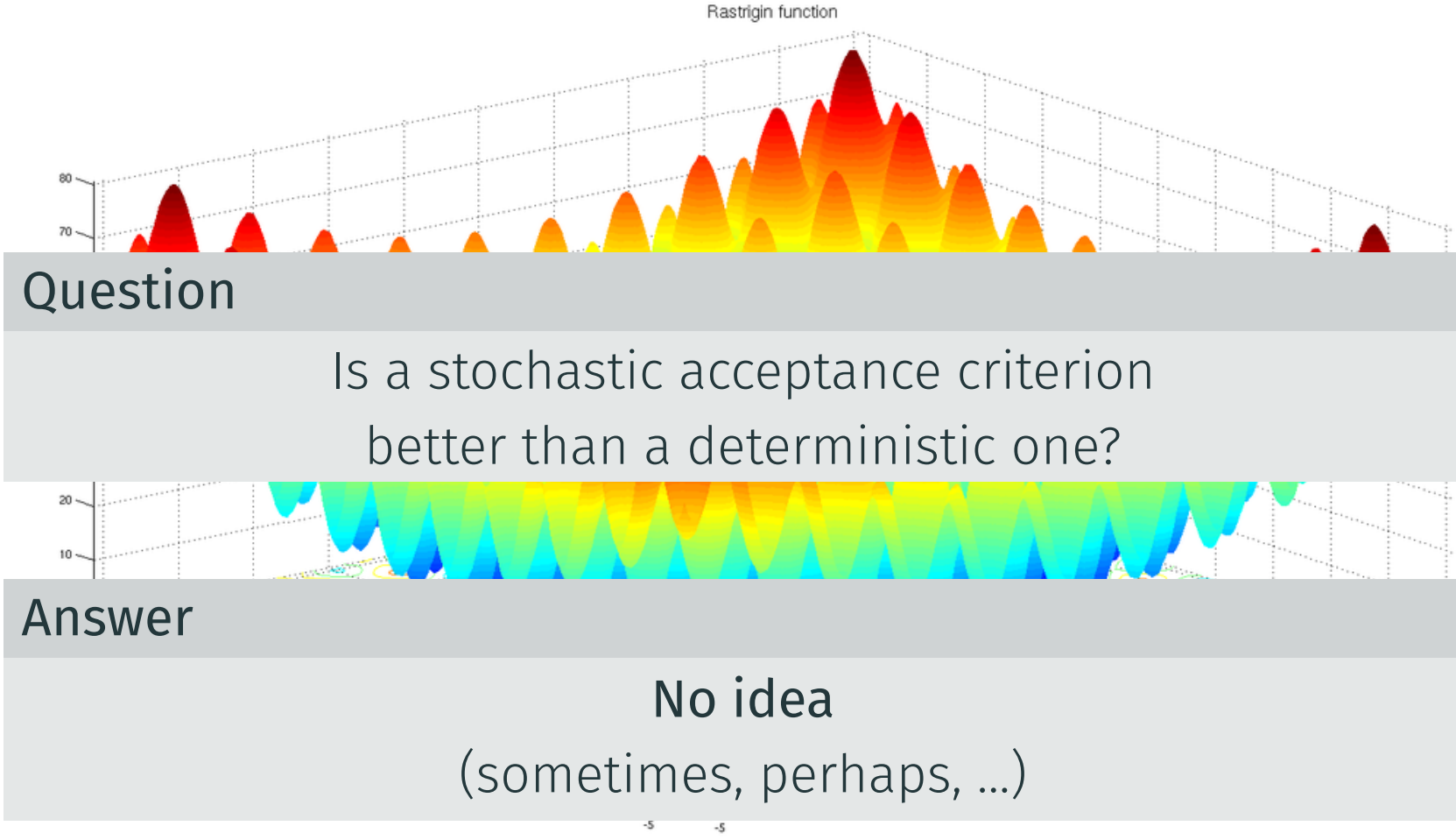


Question

Is a stochastic acceptance criterion
better than a deterministic one?

Answer

No idea



Question

Is a stochastic acceptance criterion
better than a deterministic one?

Answer

No idea
(sometimes, perhaps, ...)

Why is this type of knowledge not available in the metaheuristics literature?

Lack of knowledge in metaheuristics literature

- We do not look for it
  Focus on *novelty* and *competition*, not understanding
- We do not have the tools/methodology/protocols/standards/...

General conclusions

1. Focus on metaphors is a waste of effort

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating
   Focus should move to *understanding*

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating
   Focus should move to *understanding*
3. The review process does not allow to catch cheating

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating
   Focus should move to *understanding*
3. The review process does not allow to catch cheating
   Source code *must* be shared

Focus on “novelty”

ISA Transactions 53 (2014) 1168–1183

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Interior search algorithm (ISA): A novel approach for global optimization

Amir H. Gandomi *

The University of Akron, OH USA

A R T I C L E I N F O

Article history:
Received 8 March 2013
Received in revised form
23 October 2013
Accepted 31 March 2014
Available online 29 April 2014
This paper was recommended for publication by Prof. A.B. Rad

Keywords:
Interior search algorithm
Global optimization
Metaheuristic

A B S T R A C T

This paper presents the interior search algorithm (ISA) as a novel method for solving optimization tasks. The proposed ISA is inspired by interior design and decoration. The algorithm is different from other metaheuristic algorithms and provides new insight for global optimization. The proposed method is verified using some benchmark mathematical and engineering problems commonly used in the area of optimization. ISA results are further compared with well-known optimization algorithms. The results show that the ISA is efficiently capable of solving optimization problems. The proposed algorithm can outperform the other well-known algorithms. Further, the proposed algorithm is very simple and it only has one parameter to tune.

1. Introduction

Metaheuristic optimization algorithms are used extensively for solving complex optimization problems. Compared to conventional methods based on formal logics or mathematical programming, these metaheuristic algorithms are generally more powerful [57]. Diversification and intensification are the main features of the metaheuristic algorithms [67]. The diversification phase guarantees that the algorithm explores the search space more efficiently. The intensification phase searches through the current best solutions and selects the best candidates. Modern metaheur-

verified using some widely used benchmark problems. The results confirm the applicability of ISA for solving optimization tasks. The ISA can also outperform the existing metaheuristic algorithms. The paper is organized as follows: Section 2 provides a brief review of the metaheuristic algorithms. Section 3 presents the interior design and decoration metaphor and the characteristics of the proposed ISA, including the formulation of the algorithm. Numerical examples are presented in Section 4 to verify the efficiency of the ISA. In Section 5, the performance of the proposed algorithm is also tested using some well-known engineering design problems which have been previously employed to validate different algorithms. Finally, some concluding

Interior search algorithm

A novel metaheuristic

The proposed Interior Search Algorithm is inspired on *interior design and decoration*

Interior search algorithm

A novel metaheuristic

The proposed Interior Search Algorithm is inspired on *interior design and decoration*

- “This aesthetic process can be used for optimization by placing some mirrors near the global best(s) or fittest element(s) to find some other beautiful views.”
- “In one of these groups, called the composition group, the composition of elements is changed to find a more beautiful view.”
- “Then update each location if its fitness is improved for revival design.”

Optimization lingo?

Intelligent water drops

"The amount of soil on the edges of the iteration-best solution is reduced based on the goodness (quality) of the solution."

Optimization lingo?

Green Heron optimization algorithm

"In this case the bait helps the Green Heron bird to catch a prey and thus the solution set elements remains constant [...]."

Optimization lingo?

Cuckoo search

"The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests."

Optimization lingo?

Mine blast algorithm

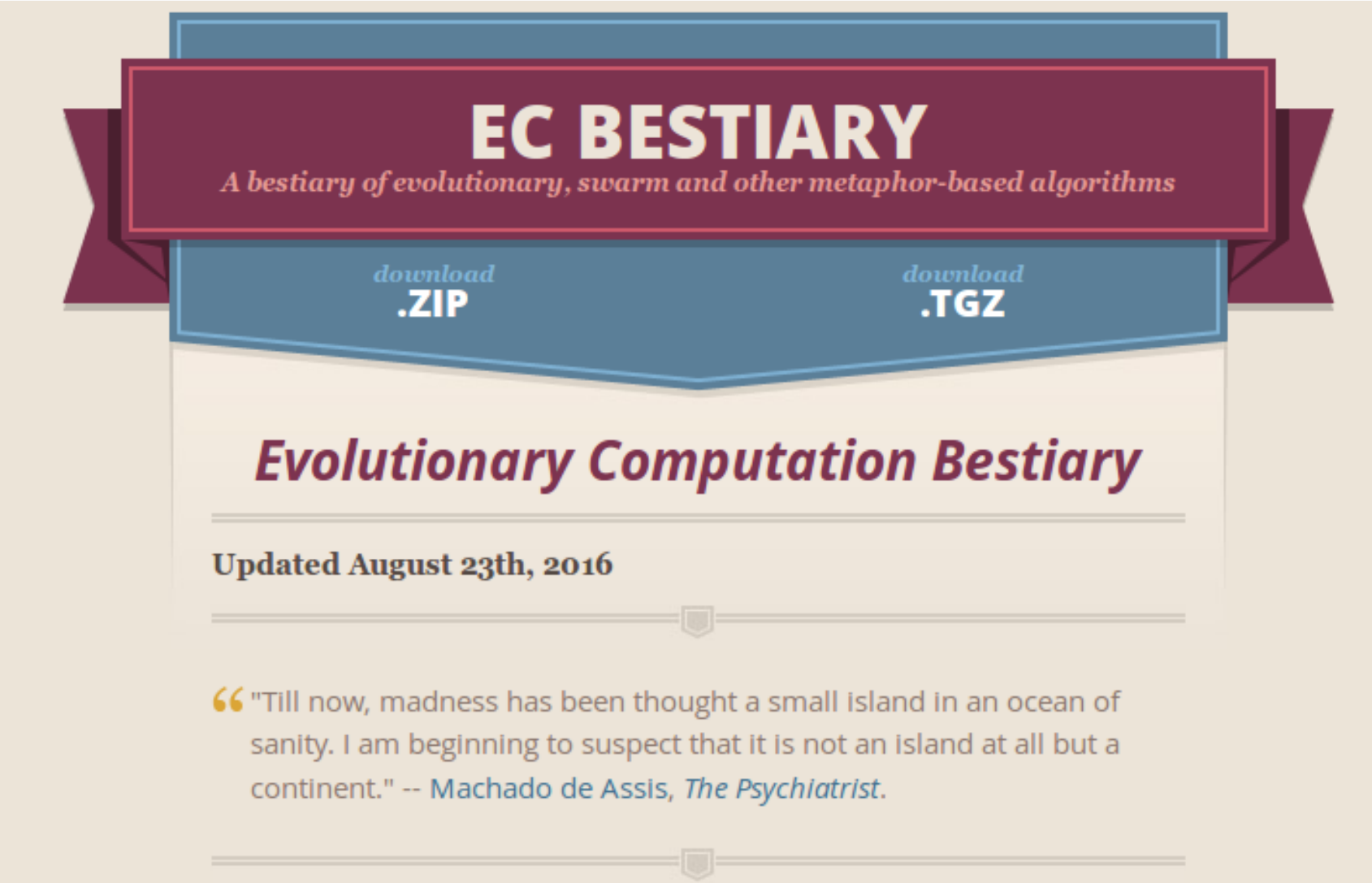
“This population is generated by a first shot explosion producing a number of individuals (shrapnel pieces). ”

Optimization lingo?

Lorentz transformation optimization

"The objective function was regarded as invariant to the reference frame, something like a transcendental entity in the space time."

http://fcampelo.github.io/EC-Bestiary/



The EC Bestiary

- African buffalo
- Anarchic societies
- Animal behavior
- Ant Lion
- Antibodies
- Bacteria
- Bacterial Chemotaxis
- Bacterial foraging
- Bacterial swarming
- Magnetotactic bacteria
- Bats
- Bees
- Bee Colonies

Ant Lion

The EC Bestiary

- Big bang
- Biogeography
- Birds mating
- Black holes
- Blind, naked mole rats
- Brainstorms
- Cats
- Central force
- Charged systems
- Chemical Reactions
- Chickens
- Clouds
- Colliding bodies

Clouds

The EC Bestiary

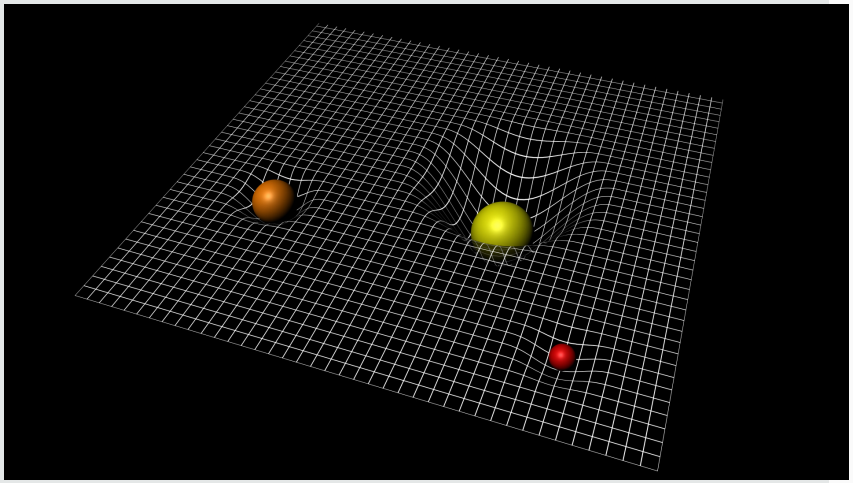
- Cuckoos
- Dolphins
- Dolphin partners
- Dolphin echolocation
- Eagles
- Ecology
- Electromagnetism
- Elephants
- Regular
- Flying
- Emotions
- FIFA World Cup
- Fireflies

Fifa World Cup

The EC Bestiary

- Cuttlefish
- Fish schools
- Fish swarms
- Flower pollination
- Fractals
- Frogs
- Frogs leaping
- Fruit Fly
- Galaxies
- Gas molecules
- Gene Expression
- General Relativity
- Glow Worms

General relativity



The EC Bestiary

- Group counselling
- Hoopoe
- Invasive weeds
- Interior design/decoration
- Ions
- Jaguars
- Krill
- Ladybirds
- Lightning
- Lion
- Locusts
- Mine blasts
- Monkeys

Mine blasts

The EC Bestiary

- Moths
- Musicians
- Optics
- Paddy fields
- Parliamentary head elections
- Penguins
- Plants
- Plants growing
- Plant propagation
- Political Imperialism
- Politicians
- Rays of light

Paddy fields

The EC Bestiary

- Salmon migrations
- Scientific method
- Sharks
- Sheep flocks
- Small World
- Spirals
- Soccer
- Social behavior
- Social Spiders
- Sports championships
- Swallows
- Symbiotic organisms
- Termites

Scientific Method

The EC Bestiary

- Wasps
- Water
- Intelligent water drops
- Water cycle
- Water evaporation
- Water flow
- Water waves
- Whales
- Wind
- Wolves
- Zombies

Zombies

The metaphor exposed paper

INTERNATIONAL TRANSACTIONS
IN OPERATIONAL RESEARCH

INTERNATIONAL
TRANSACTIONS
IN OPERATIONAL
RESEARCH

Intl. Trans. in Op. Res. 00 (2013) 1–16
DOI: 10.1111/itor.12001

Metaheuristics—the metaphor exposed

Kenneth Sörensen

University of Antwerp, Operations Research Group ANT/OR, Prinsstraat 13 – B5xx, 2000 Antwerp, Belgium
E-mail: kenneth.sorensen@ua.ac.be [Sörensen]

Received 2 November 2012; received in revised form 12 November 2012; accepted 13 November 2012

Abstract

In recent years, the field of combinatorial optimization has witnessed a true tsunami of "novel" metaheuristic methods, most of them based on a metaphor of some natural or man-made process. The behavior of virtually any species of insects, the flow of water, musicians playing together – it seems that no idea is too far-fetched to serve as inspiration to launch yet another metaheuristic. In this paper, we will argue that this line of research is threatening to lead the area of metaheuristics away from scientific rigor. We will examine the historical context that gave rise to the increasing use of metaphors as inspiration and justification for the development of new methods, discuss the reasons for the vulnerability of the metaheuristics field to this line of research, and point out its fallacies. At the same time, truly innovative research of high quality is being performed as well. We conclude the paper by discussing some of the properties of this research and by pointing out some of the most promising research avenues for the field of metaheuristics.

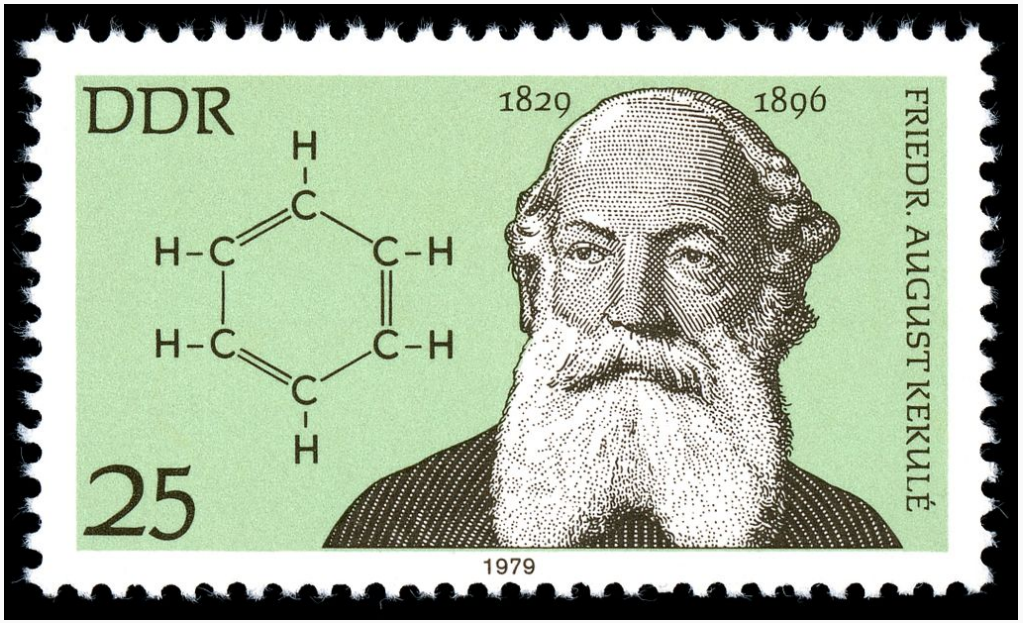
The metaphor exposed

1. Metaphor is not enough to motivate a “novel” method
2. “Novel” methods are generally not novel at all
3. “Novelty” is not a quality criterion
4. Creating a “novel” method is trivial
5. Methods should not change vocabulary on a per-method basis
6. Methods should be tested under adequate protocols
7. Knowledge gained is more important than performance

The metaphor exposed

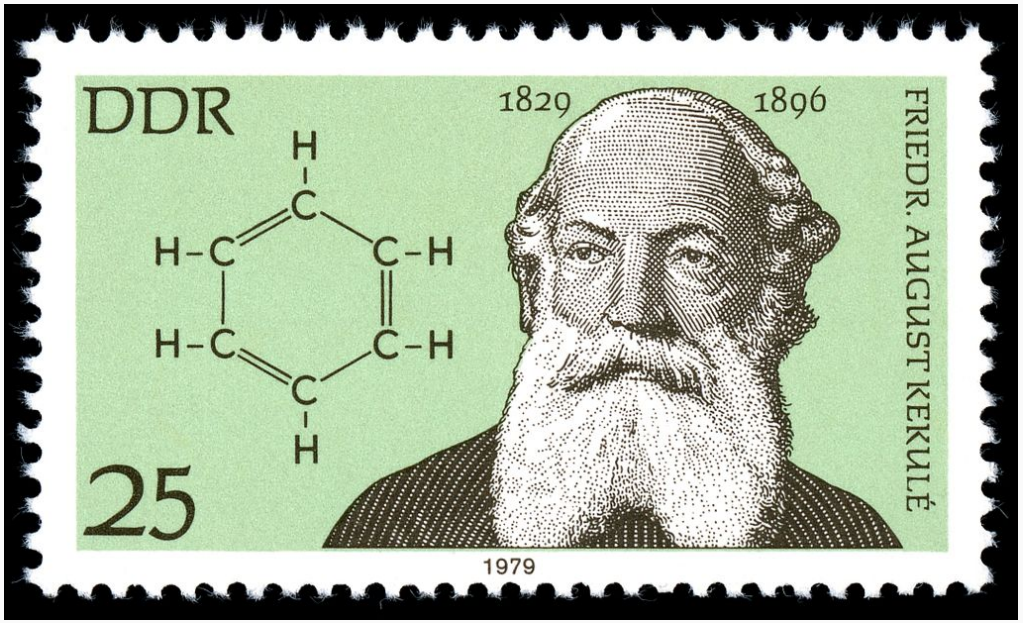
1. Metaphor is not enough to motivate a “novel” method
2. “Novel” methods are generally not novel at all
3. “Novelty” is not a quality criterion
4. Creating a “novel” method is trivial
5. Methods should not change vocabulary on a per-method basis
6. Methods should be tested under adequate protocols
7. Knowledge gained is more important than performance

What about the name?



August Kekulé discovered the structure of benzene in 1865.

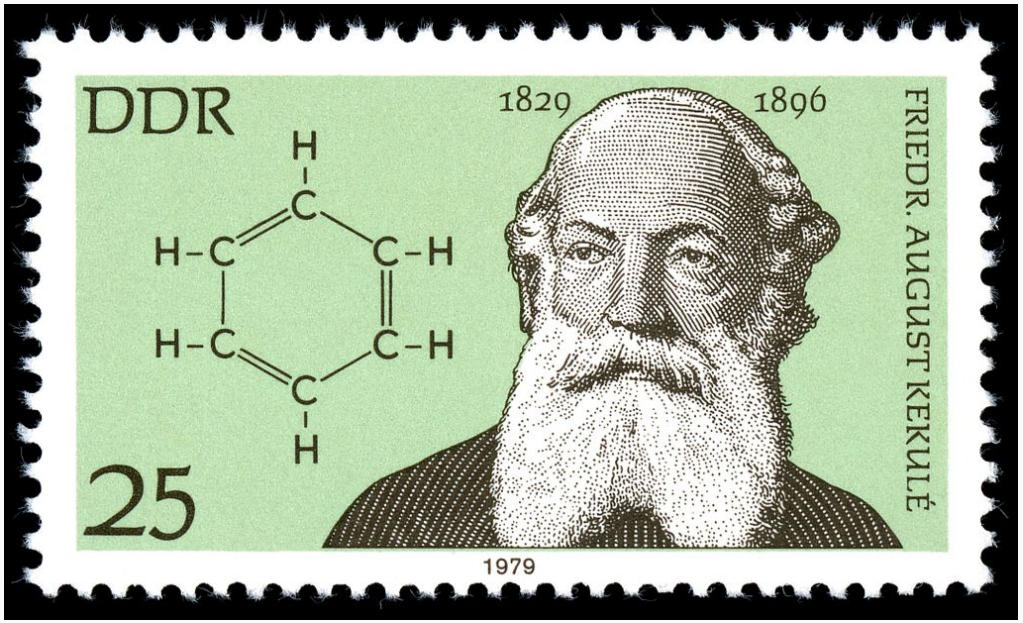
What about the name?



August Kekulé discovered the structure of benzene in 1865.

After dreaming of an **ouroboros**

What about the name?



August Kekulé discovered the structure of benzene in 1865.

After dreaming of an **ouroboros**

Metaphor

He told the world about it in 1890

Focus on competition

Are metaphors the only problem?

RESEARCH ARTICLE

ScienceAsia 38 (2012): **307–318**

doi: 10.2306/scienceasia1513-1874.2012.38.307

An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem

Tantikorn Pichpibul[a], **Ruengsak Kawtummachai**[b,*]

[a] School of Manufacturing Systems and Mechanical Engineering, Sirindhorn International Institute of Technology, Thammasat University, Pathumthani 12121 Thailand

[b] Faculty of Business Administration, Panyapiwat Institute of Management, Chaengwattana Road, Nonthaburi 11120 Thailand

*Corresponding author, e-mail: ruengsakkaw@pim.ac.th

Received 1 Aug 2011
Accepted 20 Jun 2012

ABSTRACT: In this paper, we have proposed an algorithm that has been improved from the classical Clarke and Wright savings algorithm (CW) to solve the capacitated vehicle routing problem. The main concept of our proposed algorithm is to hybridize the CW with tournament and roulette wheel selections to determine a new and efficient algorithm. The objective is to find the feasible solutions (or routes) to minimize travelling distances and number of routes. We have tested the proposed algorithm with 84 problem instances and the numerical results indicate that our algorithm outperforms CW and the optimal solution is obtained in 81% of all tested instances (68 out of 84). The average deviation between our solution and the optimal one is always very low (0.14%).

KEYWORDS: heuristics, optimization, tournament selection, roulette wheel selection

INTRODUCTION

The capacitated vehicle routing problem (CVRP) was initially introduced by Dantzig and Ramser[1] in their article on a truck dispatching problem and, consequently, became one of the most important and widely

branch-and-bound algorithm[6], a branch-and-cut algorithm[7–9], and a branch-and-cut-and-price algorithm[10]. In these algorithms, CVRP instances involving more than 100 customers can rarely be solved to optimality due to a huge amount of computation time. Second, a heuristic algorithm, which is an algorithm that

Clarke and Wright algorithm

Source

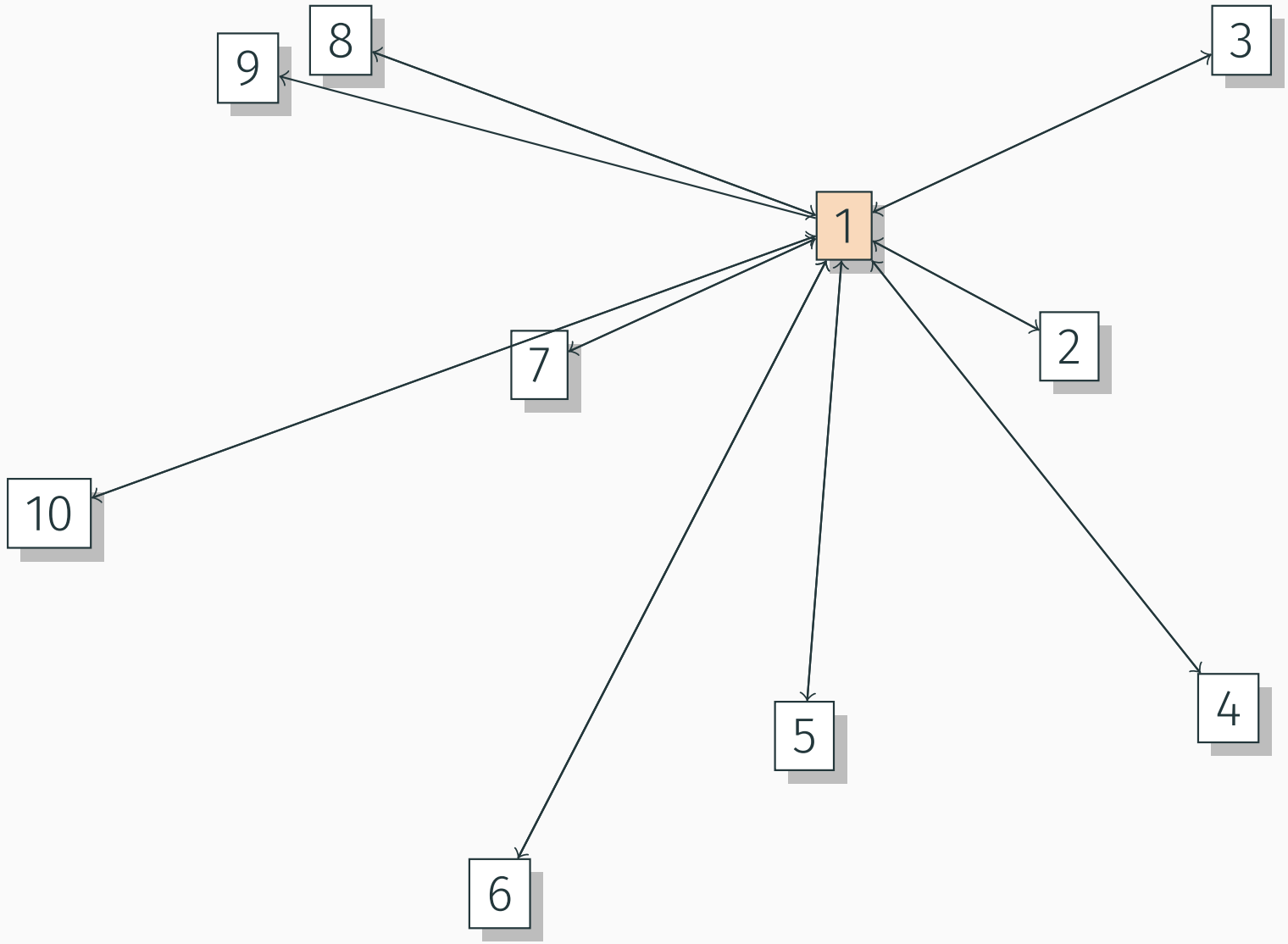
G. Clarke G and J.W. Wright, Scheduling of vehicles from a central depot to a number of delivery points, *Operations Research*, 12, 568-581, 1964.

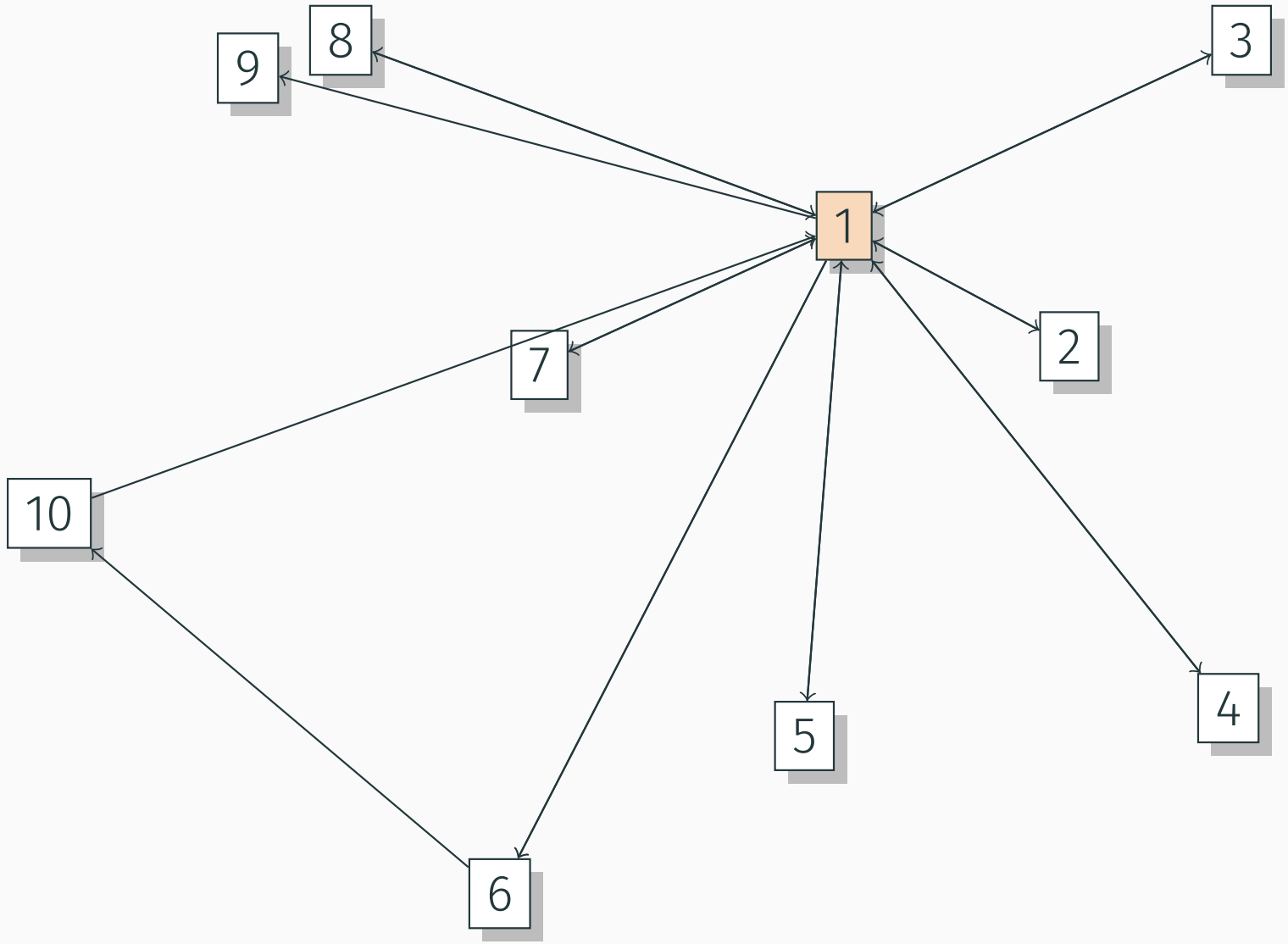
Principles

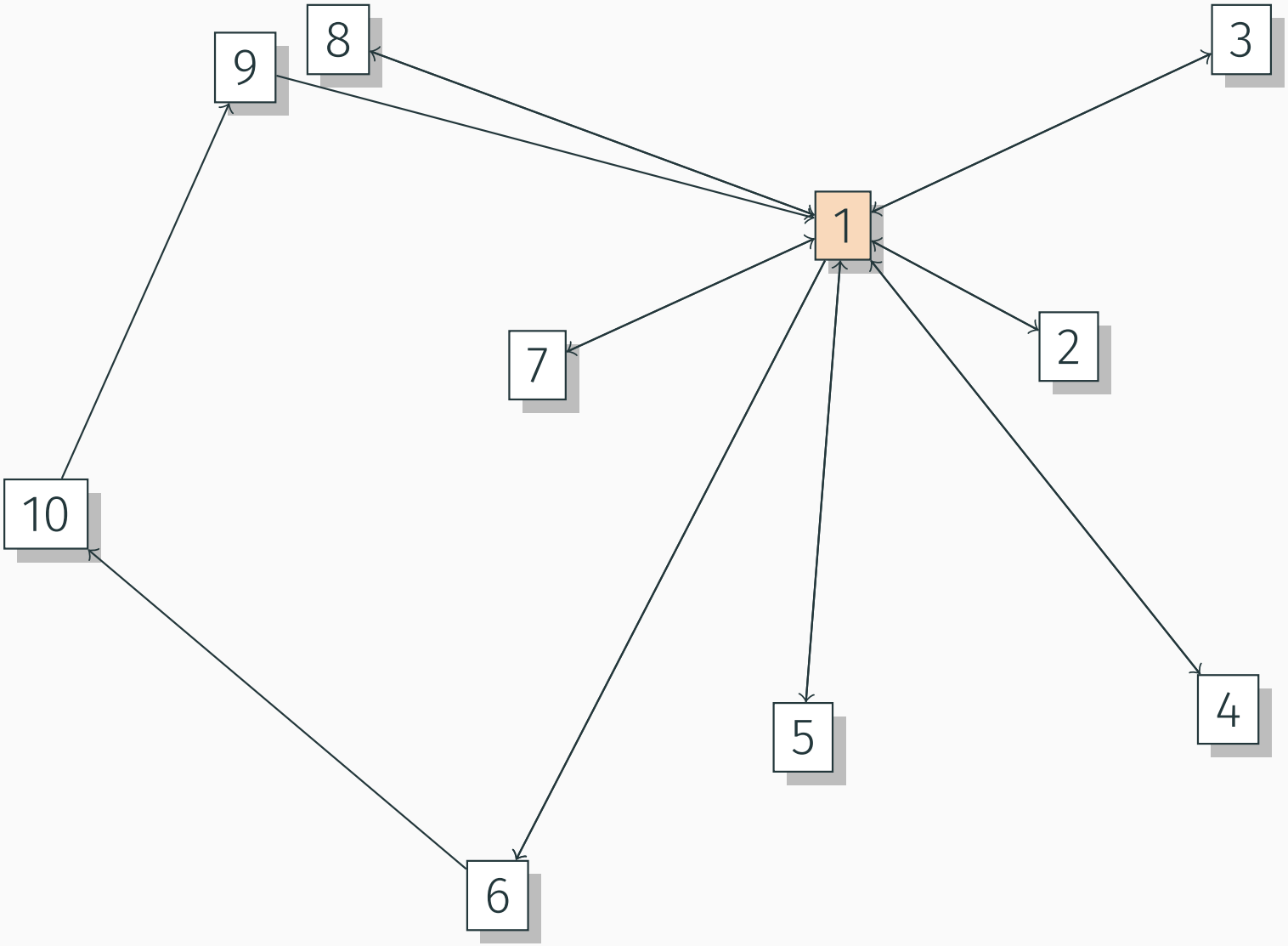
- Create a separate route per customer
- Connect routes according to the largest possible *savings*
- Repeat while routes can be connected

Saving

$s(i,j) = d(D,i) + d(D,j) - d(i,j)$


8
9
3
1
7
2
10
6
5
4


8
9
3
1
7
2
10
5
4
6


8
9
3
1
7
2
10
5
4
6

Improved Clarke and Wright

Principle

- Randomize the savings list
- “use a combination of tournament and roulette wheel selection”

Selecting a saving from the savings list:

1. Pick a random number T between 3 and 7
2. Use roulette wheel selection to select one of best T savings

“GRASP”

Improved Clarke and Wright

(i,j)	$s(i,j)$
(6,10)	86
(9,10)	83
(8,9)	78
(5,6)	77
(8,10)	66
(7,10)	57
(4,5)	55
(6,7)	50
(5,10)	49
...	...
(2,8)	0
(3,7)	0

Improved Clarke and Wright

(i,j)	$s(i,j)$
(6,10)	86
(9,10)	83
(8,9)	78
(5,6)	77
(8,10)	66
(7,10)	57
(4,5)	55
(6,7)	50
(5,10)	49
...	...
(2,8)	0
(3,7)	0

T=5

Improved Clarke and Wright

(i,j)	$s(i,j)$
(6,10)	86
(9,10)	83
(8,9)	78
(5,6)	77
(8,10)	66
(7,10)	57
(4,5)	55
(6,7)	50
(5,10)	49
...	...
(2,8)	0
(3,7)	0

T=5

Total = 86 + 83 + 78 + 77 + 66 = 390

$$p_{(6,10)} = 86/390 = .221 \qquad P = .221$$
$$p_{(9,10)} = 83/390 = .213 \qquad P = .434$$
$$p_{(8,9)} = 78/390 = .200 \qquad P = .634$$
$$p_{(5,6)} = 77/390 = .197 \qquad P = .831$$
$$p_{(8,10)} = 66/390 = .169 \qquad P = 1.000$$

Improved Clarke and Wright

(i,j)	$s(i,j)$
(6,10)	86
(9,10)	83
(8,9)	78
(5,6)	77
(8,10)	66
(7,10)	57
(4,5)	55
(6,7)	50
(5,10)	49
...	...
(2,8)	0
(3,7)	0

T=5

Total = 86 + 83 + 78 + 77 + 66 = 390

$$p_{(6,10)} = 86/390 = .221 \qquad P = .221$$
$$p_{(9,10)} = 83/390 = .213 \qquad P = .434$$
$$p_{(8,9)} = 78/390 = .200 \qquad P = .634$$
$$p_{(5,6)} = 77/390 = .197 \qquad P = .831$$
$$p_{(8,10)} = 66/390 = .169 \qquad P = 1.000$$

Random number $u \sim U[0,1]$

$u = .732 \rightarrow (5,6)$

Improved Clarke and Wright

Advantages of ICW

- Generates a different solution at every iteration (10.000 iterations)
- Very fast
- Very easy to implement
- Extremely good results

A-n32-k5	**784**	784
A-n33-k5	**661**	661
A-n33-k6	**742**	742
A-n34-k5	**778**	778
A-n36-k5	**799**	799
A-n37-k5	**669**	669
A-n37-k6	**949**	949
A-n38-k5	**730**	730
A-n39-k5	**822**	822
A-n39-k6	**831**	831
A-n44-k7	**937**	937
A-n45-k6	**944**	944
A-n45-k7	**1146**	1146
A-n46-k7	**914**	914
A-n48-k7	**1073**	1073
A-n53-k7	**1010**	1010
A-n54-k7	**1167**	1167
A-n55-k9	**1073**	1073
A-n60-k9	**1354**	1354
A-n61-k9	**1034**	1034
A-n62-k8	1298	1288
A-n63-k9	**1616**	1616
A-n63-k10	**1314**	1314
A-n64-k9	1415	1401
A-n65-k9	**1174**	1174
A-n69-k9	**1159**	1159
A-n80-k10	1772	1763

Improved Clarke and Wright

Advantages of ICW

- Generates a different solution at every iteration (10.000 iterations)
- Very fast
- Very easy to implement

- Extremely good results

  ...too good?

A-n32-k5	**784**	784
A-n33-k5	**661**	661
A-n33-k6	**742**	742
A-n34-k5	**778**	778
A-n36-k5	**799**	799
A-n37-k5	**669**	669
A-n37-k6	**949**	949
A-n38-k5	**730**	730
A-n39-k5	**822**	822
A-n39-k6	**831**	831
A-n44-k7	**937**	937
A-n45-k6	**944**	944
A-n45-k7	**1146**	1146
A-n46-k7	**914**	914
A-n48-k7	**1073**	1073
A-n53-k7	**1010**	1010
A-n54-k7	**1167**	1167
A-n55-k9	**1073**	1073
A-n60-k9	**1354**	1354
A-n61-k9	**1034**	1034
A-n62-k8	1298	1288
A-n63-k9	**1616**	1616
A-n63-k10	**1314**	1314
A-n64-k9	1415	1401
A-n65-k9	**1174**	1174
A-n69-k9	**1159**	1159
A-n80-k10	1772	1763

Improved Clarke and Wright

Advantages of ICW

- Generates a different solution at every iteration (10.000 iterations)
- Very fast
- Very easy to implement

  ...too easy?
- Extremely good results

  ...too good?

A-n32-k5	**784**	784
A-n33-k5	**661**	661
A-n33-k6	**742**	742
A-n34-k5	**778**	778
A-n36-k5	**799**	799
A-n37-k5	**669**	669
A-n37-k6	**949**	949
A-n38-k5	**730**	730
A-n39-k5	**822**	822
A-n39-k6	**831**	831
A-n44-k7	**937**	937
A-n45-k6	**944**	944
A-n45-k7	**1146**	1146
A-n46-k7	**914**	914
A-n48-k7	**1073**	1073
A-n53-k7	**1010**	1010
A-n54-k7	**1167**	1167
A-n55-k9	**1073**	1073
A-n60-k9	**1354**	1354
A-n61-k9	**1034**	1034
A-n62-k8	1298	1288
A-n63-k9	**1616**	1616
A-n63-k10	**1314**	1314
A-n64-k9	1415	1401
A-n65-k9	**1174**	1174
A-n69-k9	**1159**	1159
A-n80-k10	1772	1763

Instance	ICW	Best known
A-n32-k5	784	784
A-n33-k5	661	661
A-n33-k6	742	742
A-n34-k5	778	778
A-n36-k5	799	799
A-n37-k5	669	669
A-n37-k6	949	949
A-n38-k5	730	730
A-n39-k5	822	822
A-n39-k6	831	831
A-n44-k7	937	937
A-n45-k6	944	944
A-n45-k7	1146	1146
A-n46-k7	914	914
A-n48-k7	1073	1073
A-n53-k7	1010	1010
A-n54-k7	1167	1167
A-n55-k9	1073	1073
A-n60-k9	1354	1354
A-n61-k9	1034	1034
A-n62-k8	1298	1288
A-n63-k9	1616	1616
A-n63-k10	1314	1314
A-n64-k9	1415	1401
A-n65-k9	1174	1174
A-n69-k9	1159	1159
A-n80-k10	1772	1763

Instance	ICW	Best known	RICW* avg.	RICW* best
			20 repetitions	
A-n32-k5	784	784	793.30	784
A-n33-k5	661	661	671.05	661
A-n33-k6	742	742	744.75	742
A-n34-k5	778	778	792.30	785
A-n36-k5	799	799	805.20	805
A-n37-k5	669	669	686.25	669
A-n37-k6	949	949	965.95	949
A-n38-k5	730	730	752.75	737
A-n39-k5	822	822	838.30	825
A-n39-k6	831	831	837.30	833
A-n44-k7	937	937	955.25	947
A-n45-k6	944	944	972.55	954
A-n45-k7	1146	1146	1160.90	1153
A-n46-k7	914	914	922.40	914
A-n48-k7	1073	1073	1099.50	1097
A-n53-k7	1010	1010	1039.15	1029
A-n54-k7	1167	1167	1177.30	1172
A-n55-k9	1073	1073	1086.80	1084
A-n60-k9	1354	1354	1367.85	1362
A-n61-k9	1034	1034	1045.35	1043
A-n62-k8	1298	1288	1317.15	1310
A-n63-k9	1616	1616	1646.50	1630
A-n63-k10	1314	1314	1324.35	1315
A-n64-k9	1415	1401	1441.30	1429
A-n65-k9	1174	1174	1204.80	1193
A-n69-k9	1159	1159	1177.85	1173
A-n80-k10	1772	1763	1805.30	1789

Instance	ICW	Best known	RICW* avg.	RICW* best
			20 repetitions	
A-n32-k5	784	784	793.30	784
A-n33-k5	661	661	671.05	661
A-n33-k6	742	742	744.75	742
A-n34-k5	778	778	792.30	785
A-n36-k5	799	799	805.20	805
A-n37-k5	669	669	686.25	669
A-n37-k6	949	949	965.95	949
A-n38-k5	730	730	752.75	737
A-n39-k5	822	822	838.30	825
A-n39-k6	831	831	837.30	833
A-n44-k7	937	937	955.25	947
A-n45-k6	944	944	972.55	954
A-n45-k7	1146	1146	1160.90	1153
A-n46-k7	914	914	922.40	914
A-n48-k7	1073	1073	1099.50	1097
A-n53-k7	1010	1010	1039.15	1029
A-n54-k7	1167	1167	1177.30	1172
A-n55-k9	1073	1073	1086.80	1084
A-n60-k9	1354	1354	1367.85	1362
A-n61-k9	1034	1034	1045.35	1043
A-n62-k8	1298	1288	1317.15	1310
A-n63-k9	1616	1616	1646.50	1630
A-n63-k10	1314	1314	1324.35	1315
A-n64-k9	1415	1401	1441.30	1429
A-n65-k9	1174	1174	1204.80	1193
A-n69-k9	1159	1159	1177.85	1173
A-n80-k10	1772	1763	1805.30	1789

Instance	ICW	Best known	RICW* avg.	RICW* best
			20 repetitions	
A-n32-k5	784	784	793.30	784
A-n33-k5	661	661	671.05	661
A-n33-k6	742	742	744.75	742
A-n34-k5	778	778	792.30	785
A-n36-k5	799	799	805.20	805
A-n37-k5	669	669	686.25	669
A-n37-k6	949	949	965.95	949
A-n38-k5	730	730	752.75	737
A-n39-k5	822	822	838.30	825
A-n39-k6	831	831	837.30	833
A-n44-k7	937	937	955.25	947
A-n45-k6	944	944	972.55	954
A-n45-k7	1146	1146	1160.90	1153
A-n46-k7	914	914	922.40	914
A-n48-k7	1073	1073	1099.50	1097
A-n53-k7	1010	1010	1039.15	1029
A-n54-k7	1167	1167	1177.30	1172
A-n55-k9	1073	1073	1086.80	1084
A-n60-k9	1354	1354	1367.85	1362
A-n61-k9	1034	1034	1045.35	1043
A-n62-k8	1298	1288	1317.15	1310
A-n63-k9	1616	1616	1646.50	1630
A-n63-k10	1314	1314	1324.35	1315
A-n64-k9	1415	1401	1441.30	1429
A-n65-k9	1174	1174	1204.80	1193
A-n69-k9	1159	1159	1177.85	1173
A-n80-k10	1772	1763	1805.30	1789

* RICW = REAL ICW

Explaining the anomaly

- Programming error?
- Error in the description of the code?
- Error in the reporting of the results?

Explaining the anomaly

- Programming error?
- Error in the description of the code?
- Error in the reporting of the results?
- Most likely: deliberate misrepresentation

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

WILEY

INTERNATIONAL TRANSACTIONS IN OPERATIONAL RESEARCH

Intl. Trans. in Op. Res. 00 (2017) 1–10
DOI: 10.1111/itor.12443

A critical analysis of the "improved Clarke and Wright savings algorithm"

Kenneth Sörensen, Florian Arnold and Daniel Palhazi Cuervo

ANT/OR – Operations Research Group, Department of Engineering Management, University of Antwerp, Belgium
E-mail: kenneth.sorensen@uantwerpen.be [Sörensen]; florian.arnold@uantwerpen.be [Arnold]; daniel.palhazicuervo@uantwerpen.be [Palhazi Cuervo]

Received 16 February 2017; accepted 23 June 2017

Abstract

In their paper "An improved Clarke and Wright savings algorithm for the capacitated vehicle routing problem," published in *ScienceAsia* (38, 3, 307–318, 2012), Pichpibul and Kawtummachai developed a simple stochastic extension of the well-known Clarke and Wright savings heuristic for the capacitated vehicle routing problem. Notwithstanding the simplicity of the heuristic, which they call the "improved Clarke and Wright savings algorithm" (ICW), the reported results are among the best heuristics ever developed for this problem. Through a careful reimplementation, we demonstrate that the results published in the paper could not have been produced by the ICW heuristic. Studying the reasons how this paper could have passed the peer review process to be published in an ISI-ranked journal, we have to conclude that the necessary conditions for a thorough examination of a typical paper in the field of optimization are generally lacking. We investigate how this can be improved and come to the conclusion that disclosing source code to reviewers should become a prerequisite for publication.

Keywords: optimization; combinatorial optimization; heuristics

Reasons for “cheating”

- Good results means publication
- (If done well) almost impossible to catch
- Low risk, high gain

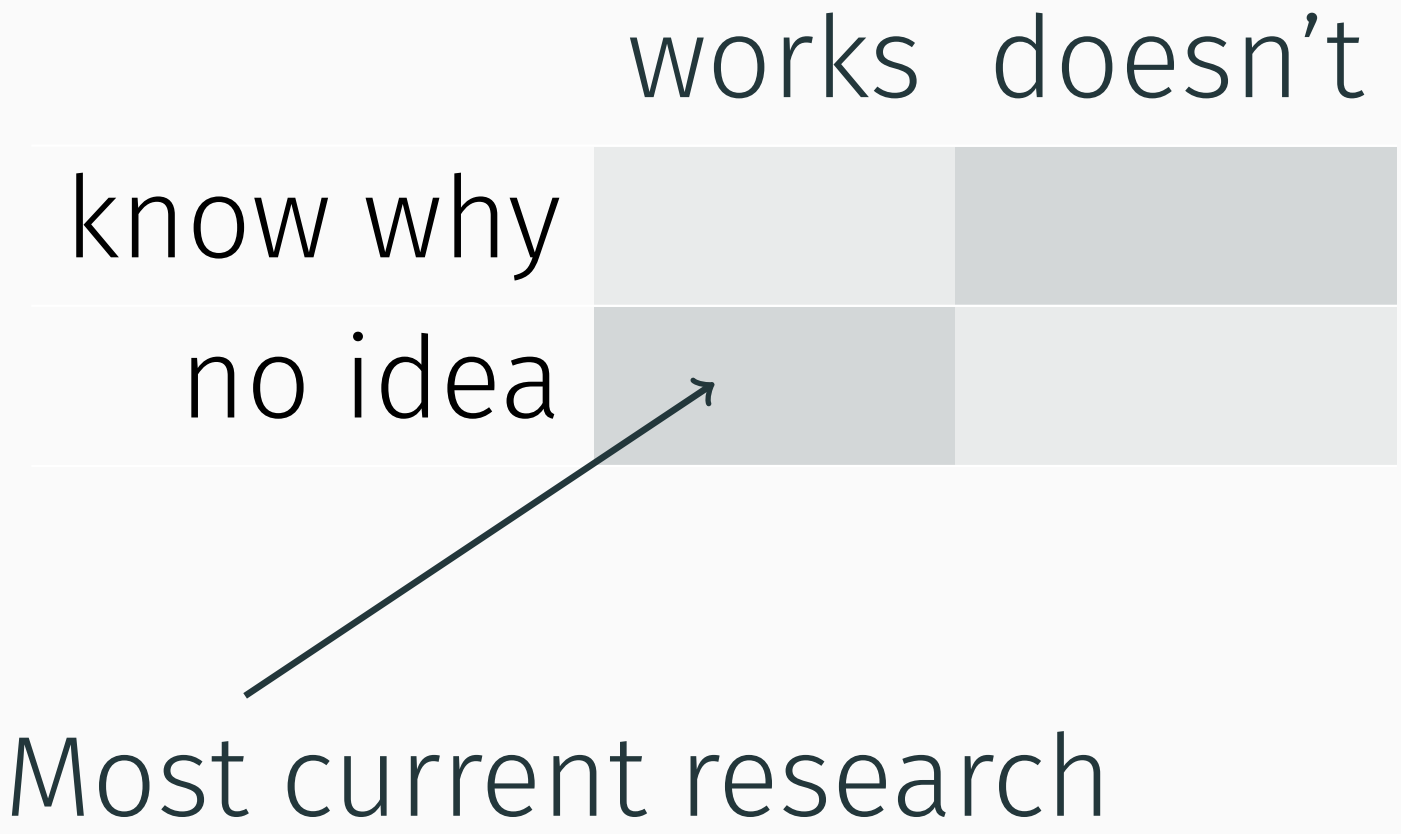
So it’s not just the metaphors

1. Focus on competition, instead of understanding (development instead of research)
2. Competition is poorly organised
3. Source code not shared

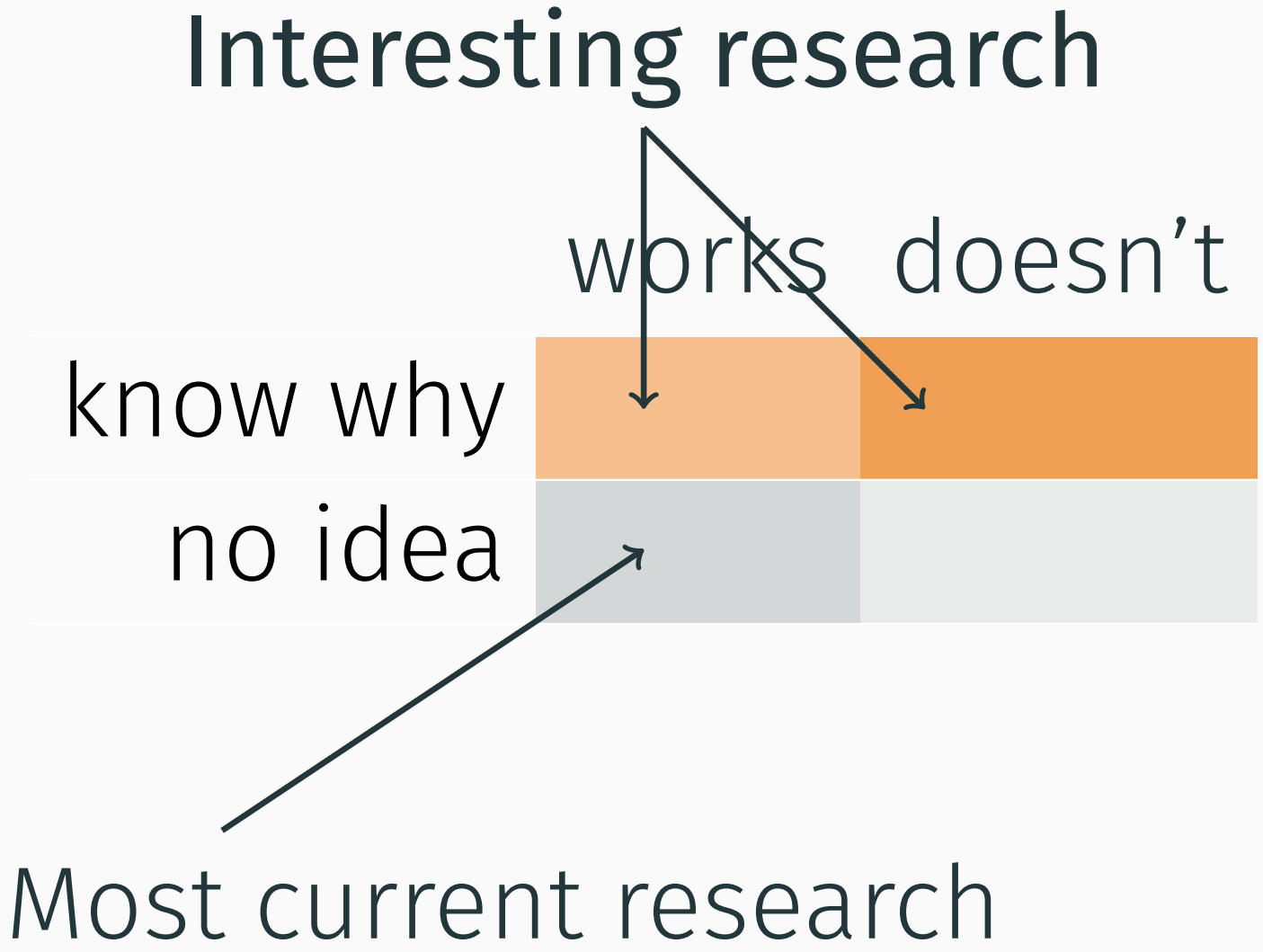
Focus on competition

	works	doesn't
know why		
no idea		

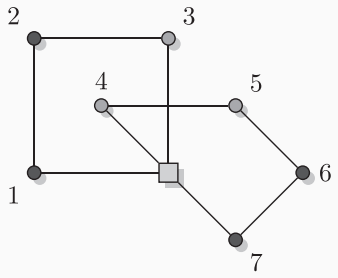
Focus on competition



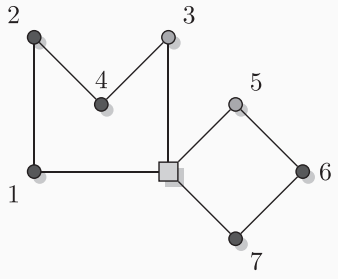
Focus on competition



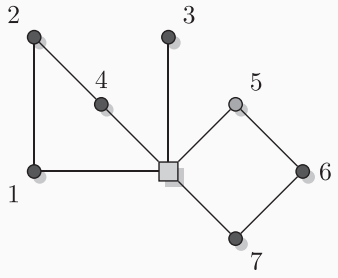
A case in point



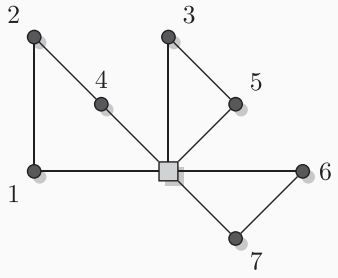
|123|4567|



|1243|567|



|124|3|567|

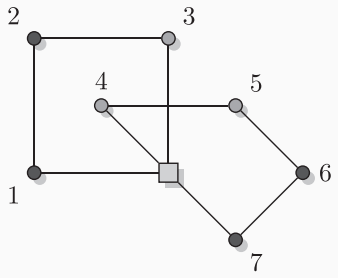


|124|35|67|

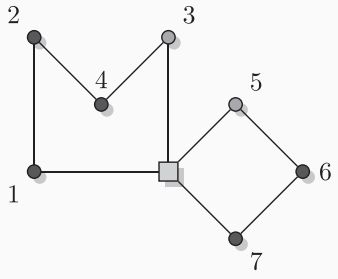
Path relinking for the VRP

- Move from VRP solution 1 to solution 2 in minimal number of moves
- True to the “spirit” of path relinking
- Extensively parametrized and tested

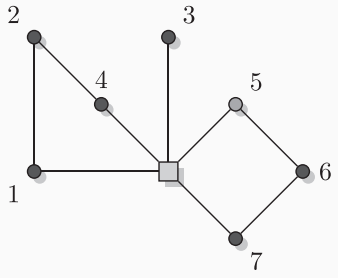
A case in point



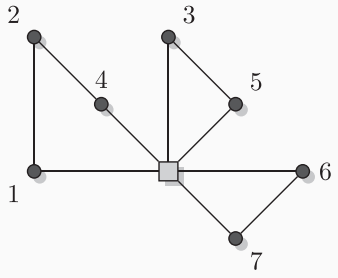
|123|4567|



|1243|567|



|124|3|567|

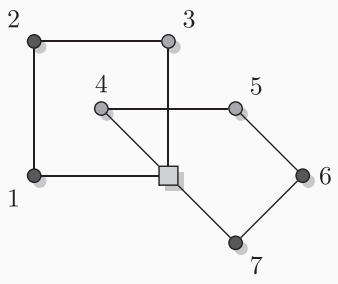


|124|35|67|

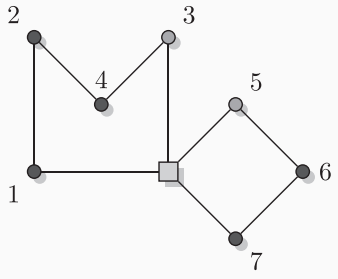
Path relinking for the VRP

- Move from VRP solution 1 to solution 2 in minimal number of moves
- True to the “spirit” of path relinking
- Extensively parametrized and tested
- **Does not work**

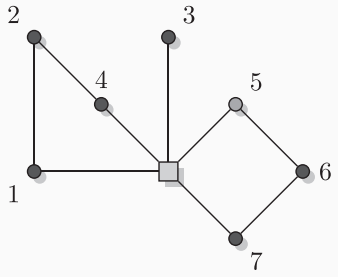
A case in point



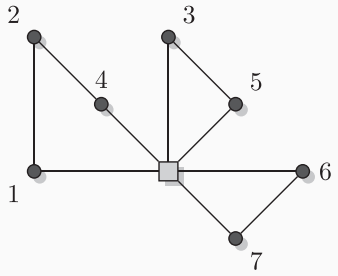
|123|4567|



|1243|567|



|124|3|567|



|124|35|67|

Path relinking for the VRP

- Move from VRP solution 1 to solution 2 in minimal number of moves
- True to the “spirit” of path relinking
- Extensively parametrized and tested
- **Does not work**

Reviewer comments

- “The authors must compare [...] with previous methods.”
- “[...] the results should be superior (or equivalent) to them.”

Computers & Operations Research 40 (2013) 3197–3205

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Statistical analysis of distance-based path relinking for the capacitated vehicle routing problem

Kenneth Sörensen *, Patrick Schittekat

University of Antwerp, Faculty of Applied Economics, ANT/OR University of Antwerp Operations Research Group, Prinsstraat 13, 2000 Antwerp, Belgium

ARTICLE INFO

Available online 13 February 2013

Keywords:
Grasp
Path relinking
Relocate distance
Edit distance
Capacitated vehicle routing problem
Statistical analysis

ABSTRACT

In this paper we develop an intelligent path relinking procedure for the capacitated vehicle routing problem, based on the relocate distance. This procedure transforms an incumbent solution into a guiding solution in a minimal number of relocate moves. In each step of the path relinking procedure, one customer is removed from the solution and re-inserted in another position.

The path relinking procedure is integrated in a GRASP (greedy randomized adaptive search procedure) and VND (variable neighborhood descent) framework and thoroughly tested. This analysis shows that the path relinking procedure is not able to improve the performance of a simple GRASP + VND metaheuristic, but some interesting conclusions can nonetheless be drawn.

A second contribution of this paper is an analysis of the computational results based on sound statistical techniques. Such an analysis can be useful for the field of metaheuristics, where computational results are generally analyzed in an ad hoc way and often with dubious statistical validity.

1. Path relinking and the vehicle routing problem

The capacitated vehicle routing problem (CVRP) is defined on a graph $G=(V,E)$ with $V=v_0\cup\{v_1,\ldots,v_N\}$. The set $\{v_1,\ldots,v_N\}$ represents a set of customers and v_0 represents a depot. The edges represent travel costs between customers. Each of the N customer has a non-negative known demand q_i $(i=1,\ldots,N)$. This demand must be serviced by a homogeneous set of vehicles, all having capacity Q. Travel costs c_{ij} between customers i and j are known and constant. Sometimes an extra cost (called drop cost e_i) is incurred for each customer visited.

The CVRP is undoubtedly one of the best-studied problems in operations research. Given the fact that the CVRP is NP-hard, only small instances can be solved to optimality [3]. Heuristics and metaheuristics are therefore often used to solve the CVRP and a large number of metaheuristics have been developed for solving this problem. A recent overview on different metaheuristics for the CVRP can be found in Szeto et al. [22]. Notable methods include the adaptive memory search procedure of Rochat and Taillard [19], that (after more than 15 years) still tops the list of best-performing approaches on the standard Christophides

Testing Heuristics: We Have It All Wrong

J. N. Hooker
Graduate School of Industrial Administration
Carnegie Mellon University
Pittsburgh, PA 15213 USA

May 1995

Abstract

The competitive nature of most algorithmic experimentation is a source of problems that are all too familiar to the research community. It is hard to make fair comparisons between algorithms and to assemble realistic test problems. Competitive testing tells us which algorithm is faster but not why. Because it requires polished code, it consumes time and energy that could be spent doing more experiments. This paper argues that a more scientific approach of controlled experimentation, similar to that used in other empirical sciences, avoids or alleviates these problems. We have confused research and development; competitive testing is suited only for the latter.

Most experimental studies of heuristic algorithms resemble track meets more than scientific endeavors.

Typically an investigator has a bright idea for a new algorithm and wants to show that it works better, in some sense, than known algorithms. This requires computational tests, perhaps on a standard set of benchmark

Competitive testing

- Fallacy: good algorithm = good science
- Competitive testing tells us which algorithm is best, but not *why*
- Not fair: depends on coding skills, optimizations, parameter tuning, computers, compilers, ...
- Wastes enormous amounts of research time on "development"
- No standard testing protocol, window dressing is too easy
- Focus on "standard" benchmark instances leads to overfitting
- Yields very little knowledge

Lack of established testing methods

1. We found α new best-known solutions
2. We found more best/optimal solutions than method X
3. Our algorithm performs β% better on average than method X
4. We ran our algorithm γ times and report the best solution value over all runs

- What about statistical significance?

Lack of established testing methods

1. We found α new best-known solutions
2. We found more best/optimal solutions than method X
3. Our algorithm performs β% better on average than method X
4. We ran our algorithm γ times and report the best solution value over all runs

- What about statistical significance?

Solution

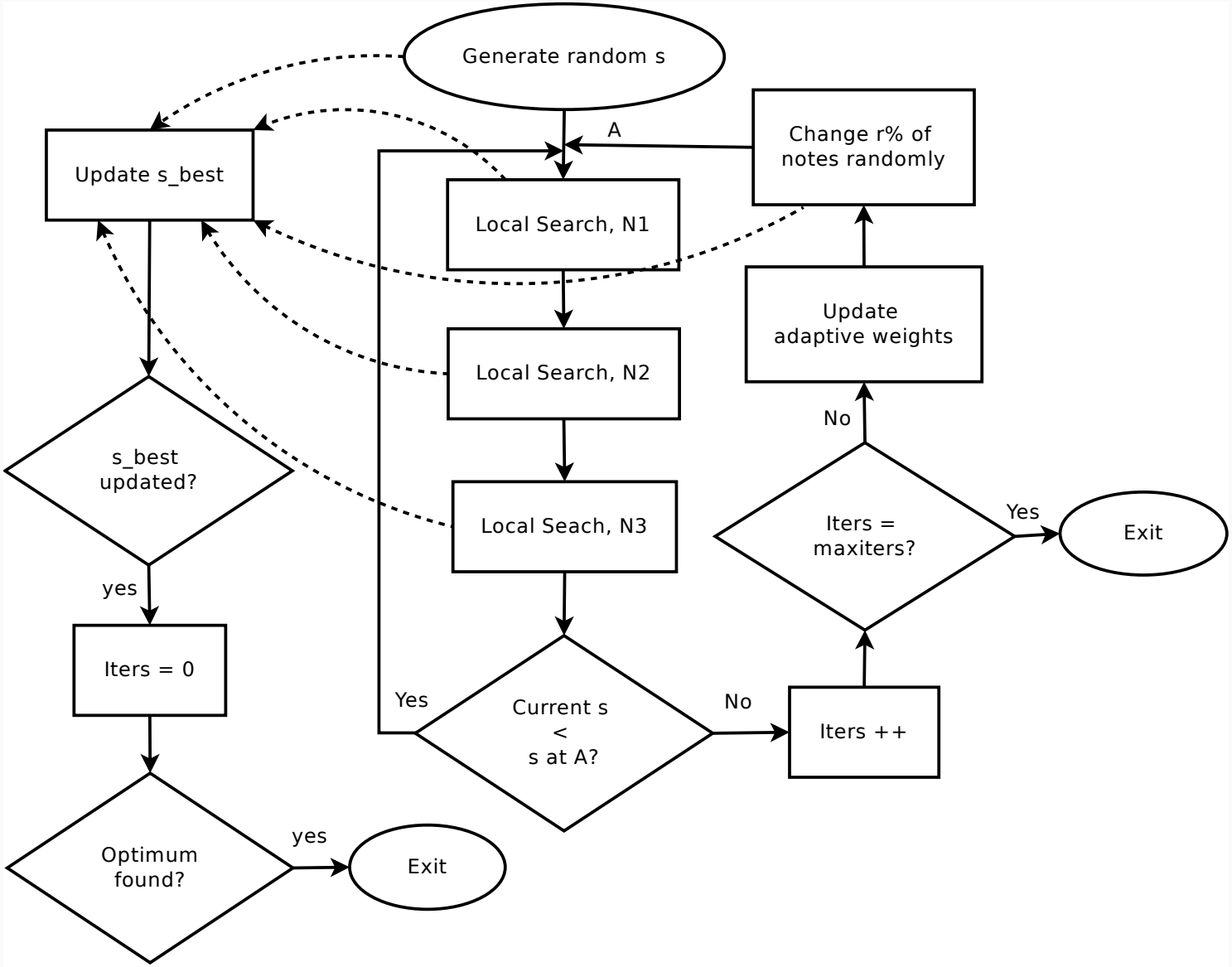
Statistical tests are available for this!

Focus on understanding

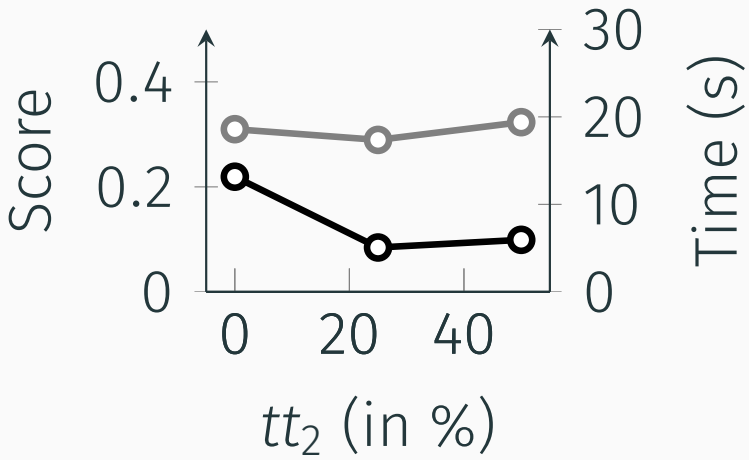
Focus on understanding: some ideas

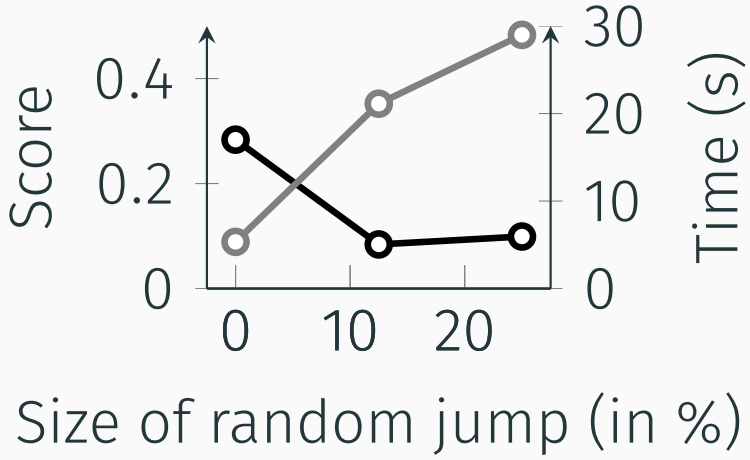
1. Decomposition of algorithms + statistical experiments
2. Allow negative results if insightful
3. Lessons from other sciences
    - Controlled experimentation
    - Structured reviews
    - Meta-analyses

Deconstruction



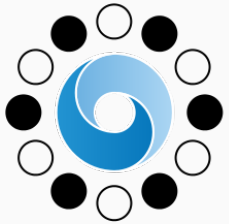
Deconstruction





Fixed instance sets

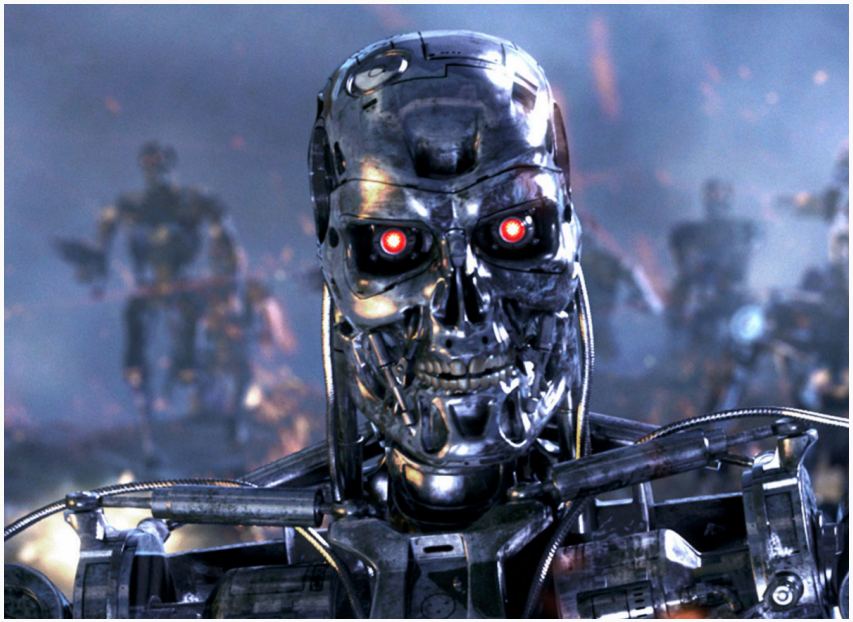
Fixed instance sets



AlphaGo

Fixed instance sets

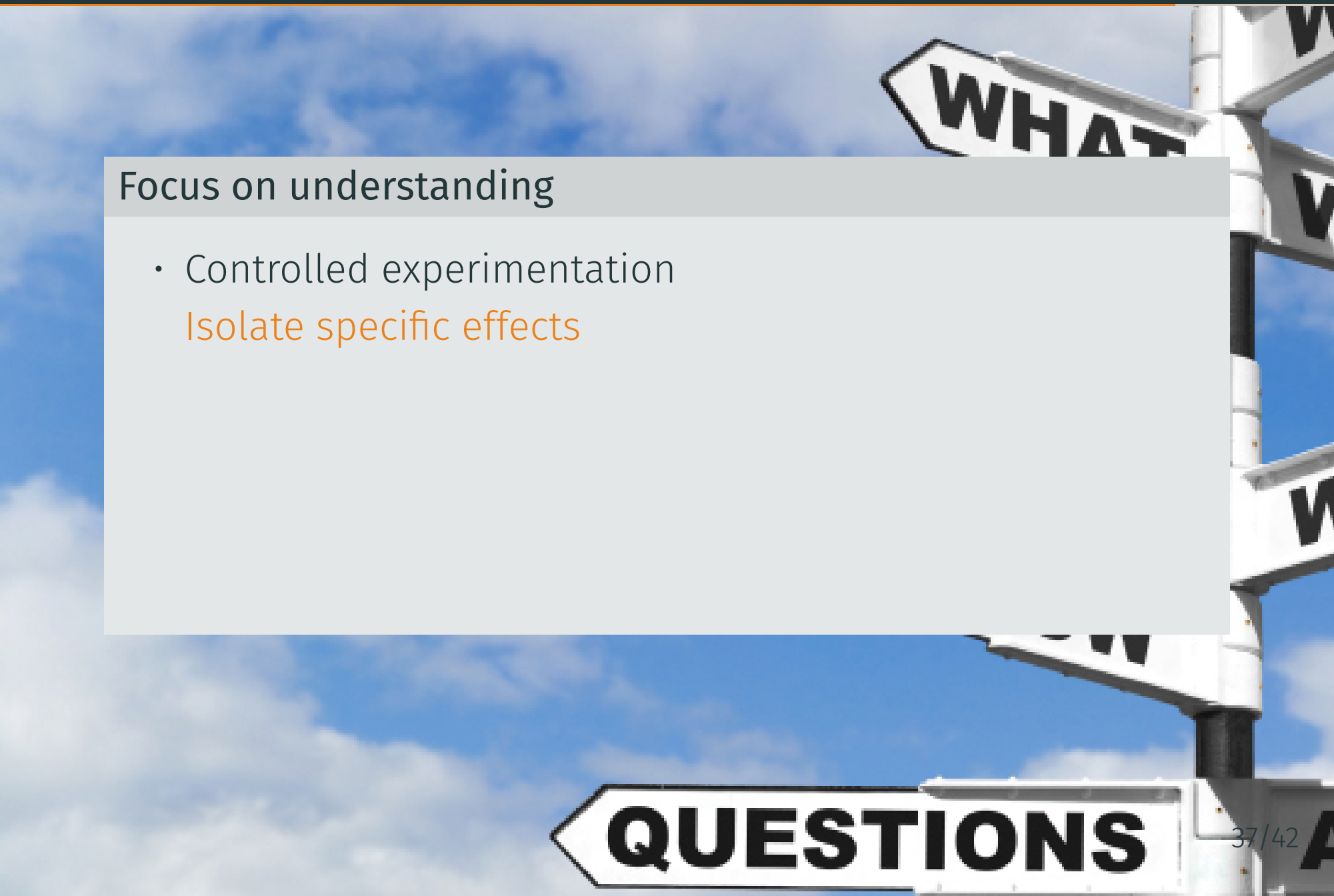
AlphaGo



Solution?

Multi-level experimental design with randomly generated instances

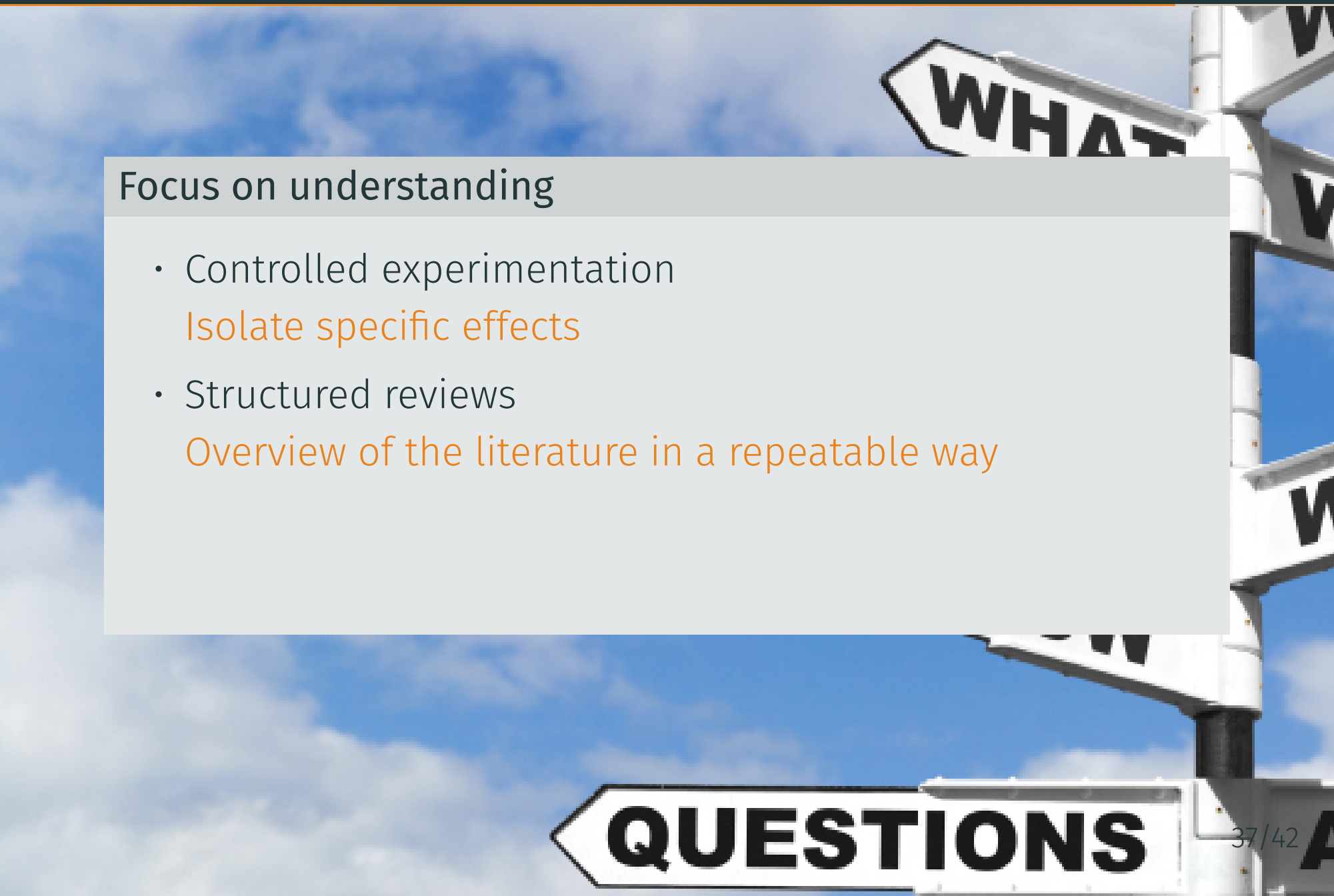
	Problem Characteristics			**Algorithmic Parameters**		
i	X_1	X_2	X_3	Z_1	Z_2	Z_3
1	0.5	10	0	0.2	12	0
2	0.5	10	0	0.4	5	1
3	0.5	10	0	0.9	15	1
4	0.5	10	0	0.9	6	0
5	1.2	8	1	0.6	10	1
6	1.2	8	1	0.1	1	1
7	1.2	8	1	0.5	5	1
8	1.2	8	1	0.2	3	0
...	...	...	...	...	...	...



Focus on understanding

- Controlled experimentation
  Isolate specific effects

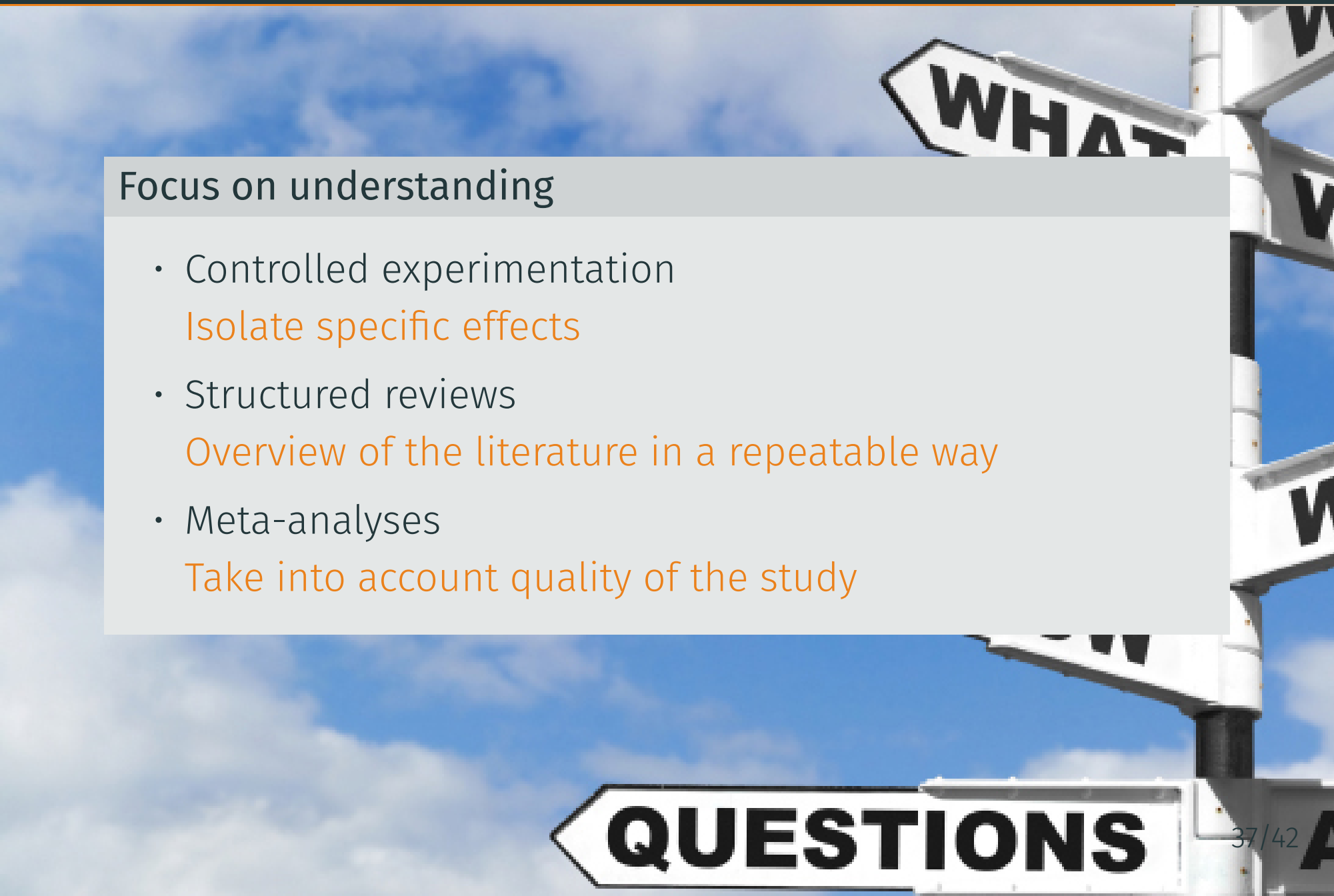
Structured review and meta-analysis



Focus on understanding

- Controlled experimentation
  Isolate specific effects
- Structured reviews
  Overview of the literature in a repeatable way

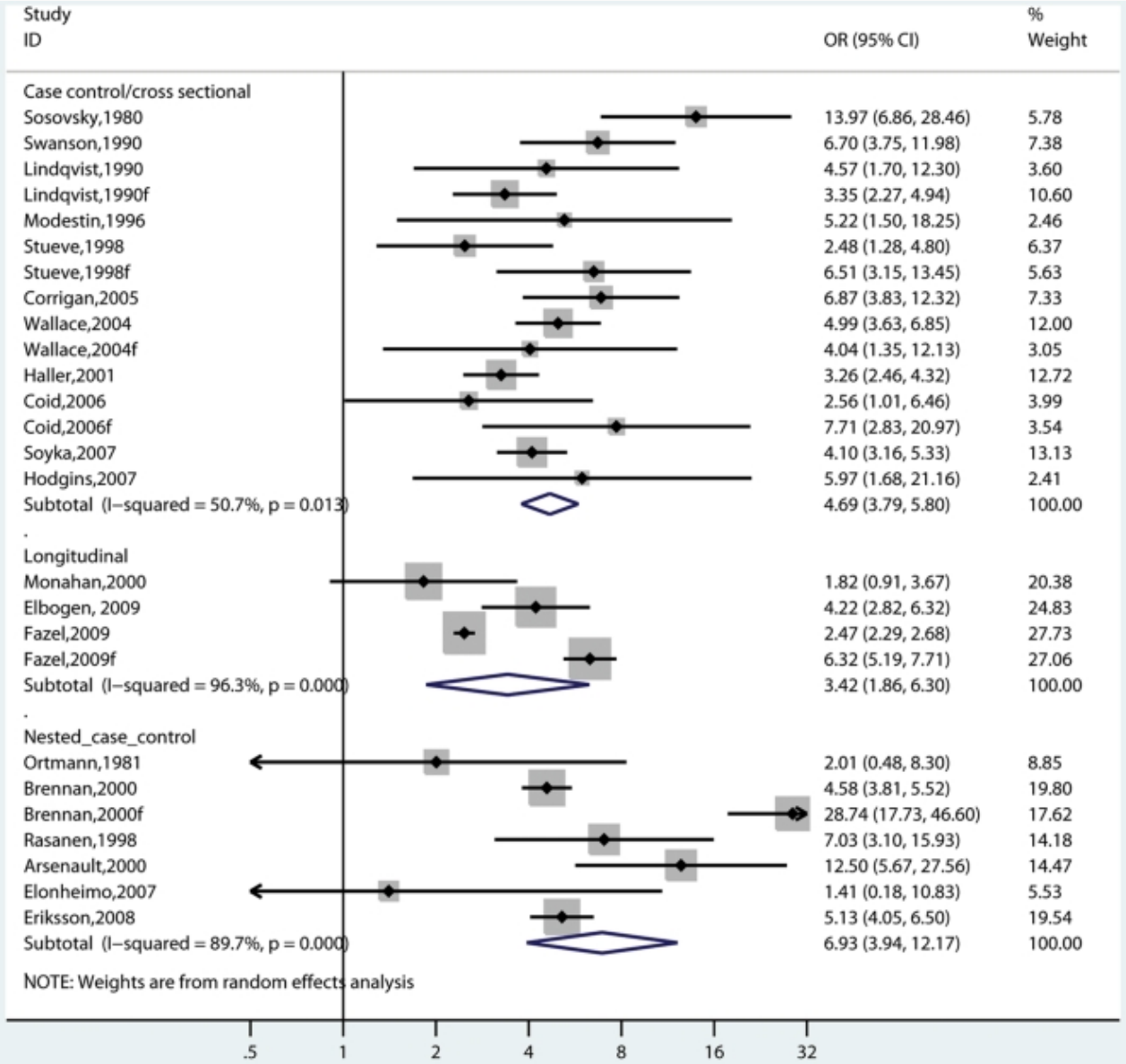
Structured review and meta-analysis



Focus on understanding

- Controlled experimentation
  Isolate specific effects
- Structured reviews
  Overview of the literature in a repeatable way
- Meta-analyses
  Take into account quality of the study

Structured reviews and meta-analysis



Release source code

Release source code

- To reviewers (+ compilation instructions)
- To the world

PERSPECTIVE

doi:10.1038/nature10836

The case for open computer programs

Darrel C. Ince[1], Leslie Hatton[2] & John Graham-Cumming[3]

Scientific communication relies on evidence that cannot be entirely included in publications, but the rise of computational science has added a new layer of inaccessibility. Although it is now accepted that data should be made available on request, the current regulations regarding the availability of software are inconsistent. We argue that, with some exceptions, anything less than the release of source programs is intolerable for results that depend on computation. The vagaries of hardware, software and natural language will always ensure that exact reproducibility remains uncertain, but withholding code increases the chances that efforts to reproduce results will fail.

The rise of computational science has led to unprecedented opportunities for scientific advance. Ever more powerful computers enable theories to be investigated that were thought almost intractable a decade ago, robust hardware technologies allow data collection in the most inhospitable environments, more data are collected, and an increasingly rich set of software tools are now available with which to analyse computer-generated data.

However, there is the difficulty of reproducibility, by which we mean the reproduction of a scientific paper's central finding, rather than exact replication of each specific numerical result down to several decimal places. We examine the problem of reproducibility (for an early attempt at solving it, see ref. 1) in the context of openly available computer programs, or code. Our view is that we have reached the point that, with some exceptions, anything less than release of actual source code is an indefensible approach for any scientific results that depend on computation, because not releasing such code raises needless, and needlessly confusing, roadblocks to reproducibility.

At present, debate rages on the need to release computer programs associated with scientific experiments[2–4], with policies still ranging from mandatory total release to the release only of natural language descriptions, that is, written descriptions of computer program algorithms. Some journals have already changed their policies on computer program openness; *Science*, for example, now includes code in the list of items that should be supplied by an author[5]. Other journals promoting code

rerunning of the code on, say, a different combination of hardware and systems software, to detect the sort of numerical computation[11,12] and interpretation[13] problems found in programming languages, which we discuss later. Without code, direct reproducibility is impossible. Indirect reproducibility refers to independent efforts to validate something other than the entire code package, for example a subset of equations or a particular code module. Here, before time-consuming reprogramming of an entire model, researchers may simply want to check that incorrect coding of previously published equations has not invalidated a paper's result, to extract and check detailed assumptions, or to run their own code against the original to check for statistical validity and explain any discrepancies.

Any debate over the difficulties of reproducibility (which, as we will show, are non-trivial) must of course be tempered by recognizing the undeniable benefits afforded by the explosion of internet facilities and the rapid increase in raw computational speed and data-handling capability that has occurred as a result of major advances in computer technology[14]. Such advances have presented science with a great opportunity to address problems that would have been intractable in even the recent past. It is our view, however, that the debate over code release should be resolved as soon as possible to benefit fully from our novel technical capabilities. On their own, finer computational grids, longer and more complex computations and larger data sets—although highly attractive to scientific researchers—do not resolve underlying computational uncertainties of proven intransigence and may even exacerbate them.

General conclusions

1. Focus on metaphors is a waste of effort

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating
   Focus should move to *understanding*

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating
   Focus should move to *understanding*
3. The review process does not allow to catch cheating

General conclusions

1. Focus on metaphors is a waste of effort
   But the entire field should improve its standards
2. Focus on competitive testing results in poor research and motivates cheating
   Focus should move to *understanding*
3. The review process does not allow to catch cheating
   Source code *must* be shared

http://antor.uantwerpen.be

kenneth.sorensen@uantwerpen.be

University of Antwerp
Operations Research Group
ANT/OR